

AR100 SERIES INTERNET ROUTER

REFERENCE MANUAL

CONTENTS ➤

COMMANDS ➤

INDEX ➤

ABOUT ➤

AR100 Series Internet Router Reference Manual
Document Number C613-03017-00 REV A.

Copyright © 1999-2000 Allied Telesyn International, Corp.
960 Stewart Drive Suite B, Sunnyvale CA 94086, USA.

All rights reserved. No part of this publication may be reproduced without prior written permission from Allied Telesyn.

Allied Telesyn International, Corp. reserves the right to make changes in specifications and other information contained in this document without prior written notice. The information provided herein is subject to change without notice. In no event shall Allied Telesyn be liable for any incidental, special, indirect, or consequential damages whatsoever, including but not limited to lost profits, arising out of or related to this manual or the information contained herein, even if Allied Telesyn has been advised of, known, or should have known, the possibility of such damages.

Contents

Preface

Purpose of this Manual	xxix
Intended Audience	xxx
Structure of this Manual	xxx
Related Manuals	xxxi
Other Documents	xxxi
Supported Standards and Protocols	xxxi
Obtaining Copies of Internet Protocols and Standards	xxxiii
Background Reading	xxxiii
Publicly Accessible Documents	xxxiv
Conventions	xxxv
Allied Telesyn Offices and Locations	xxxvii
Safety and Statutory Information	xxxviii
Safety and Statutory Statements	xxxviii
Environmental Conditions	xli
Reader's Comments	xlili

Command Summary

CHAPTER 1 Operation

Introduction	1-3
The Command Processor	1-3
User Privilege Levels	1-3
Entering Commands	1-4
Aliases	1-5
Online Help	1-5
Storing and Retrieving Configuration Information	1-6
User Authentication Facility	1-7
The User Authentication Database	1-8
Asynchronous Port Security	1-11
Telnetting from the Router	1-11
Counters	1-11
Semipermanent Manager Port	1-12
Remote Management	1-12
Monitoring and Fault Diagnosis	1-13
Event Logging	1-13
Restarts	1-13
CPU Utilisation	1-13
Memory	1-13
FLASH Memory	1-14
Physical Characteristics	1-14
The File Subsystem	1-15
File Naming Conventions	1-15

Using Wildcards to Specify Groups of Files	1-16
Working With Files	1-16
FLASH File System	1-16
Working with FFS Files	1-17
Compaction	1-17
FFS Messages	1-18
The Built-in Editor	1-18
HTTP Client and Server	1-19
Resolving Uniform Resource Locators (URLs)	1-20
Software Releases	1-20
Releases	1-20
Router Startup Operations	1-21
Downloading Releases into the Router	1-22
Install Information	1-23
Examples	1-24
Command Reference	1-26
ACTIVATE FLASH COMPACTION	1-26
ADD ALIAS	1-28
ADD USER	1-28
CLEAR FLASH TOTALLY	1-29
CREATE CONFIG	1-30
CREATE FFILE	1-31
DELETE ALIAS	1-31
DELETE FFILE	1-32
DELETE FILE	1-32
DELETE INSTALL	1-33
DELETE USER	1-33
DISABLE HTTP DEBUG	1-34
DISABLE HTTP SERVER	1-34
DISABLE USER	1-35
DUMP	1-35
EDIT	1-37
ENABLE HTTP DEBUG	1-40
ENABLE HTTP SERVER	1-40
ENABLE USER	1-41
HELP	1-41
LOAD	1-42
LOGIN	1-44
LOGOFF	1-45
MODIFY	1-45
PURGE USER	1-46
RENAME	1-46
RESET HTTP SERVER	1-47
RESET LOADER	1-47
RESET USER	1-48
RESTART	1-49
SET CONFIG	1-49
SET HELP	1-50
SET INSTALL	1-50
SET LOADER	1-51
SET MANAGER PORT	1-53
SET PASSWORD	1-54
SET SYSTEM CONTACT	1-54
SET SYSTEM LOCATION	1-55
SET SYSTEM NAME	1-55
SET SYSTEM TERRITORY	1-56
SET TIME	1-56
SET USER	1-57
SHOW ALIAS	1-58

SHOW BUFFER	1-59
SHOW CONFIG	1-61
SHOW CPU	1-63
SHOW DEBUG	1-63
SHOW EXCEPTION	1-64
SHOW FFILE	1-65
SHOW FILE	1-67
SHOW FLASH	1-67
SHOW FLASH PHYSICAL	1-69
SHOW HTTP CLIENT	1-69
SHOW HTTP DEBUG	1-70
SHOW HTTP SESSION	1-71
SHOW HTTP SERVER	1-72
SHOW INSTALL	1-74
SHOW LOADER	1-75
SHOW MANAGER PORT	1-77
SHOW STARTUP	1-77
SHOW SYSTEM	1-78
SHOW TIME	1-80
SHOW USER	1-80
UPLOAD	1-82

CHAPTER 2 Interfaces

Introduction	2-2
Naming interfaces	2-2
Ethernet	2-3
Encapsulations	2-4
Configuration	2-5
Asynchronous Interfaces	2-7
Encapsulations	2-7
Configuration	2-7
Autobauding	2-11
Displaying Interfaces	2-11
Interface Link Traps	2-12
Managing Interfaces with SNMP	2-12
Command Reference	2-13
DISABLE INTERFACE LINKTRAP	2-13
DISABLE PORT	2-14
ENABLE INTERFACE LINKTRAP	2-14
ENABLE PORT	2-15
PURGE PORT	2-15
RESET ETH	2-16
RESET ETH COUNTERS	2-16
RESET PORT	2-17
RESET PORT COUNTERS	2-17
RESET PORT HISTORY	2-18
SET INTERFACE TRAPLIMIT	2-18
SET PORT	2-19
SHOW ETH CONFIGURATION	2-22
SHOW ETH COUNTERS	2-23
SHOW ETH MACADDRESS	2-29
SHOW ETH RECEIVE	2-29
SHOW INTERFACE	2-30
SHOW PORT	2-33

CHAPTER 3 Point-to-Point Protocol (PPP)

Introduction	3-3
The Point-to-Point Protocol	3-3
Encapsulation	3-3
Control Protocols	3-4
LCP Options	3-5
Link Quality Management	3-6
Multilink PPP	3-6
Bandwidth Allocation Protocol	3-7
Dial-On-Demand	3-8
Bandwidth on Demand	3-8
PPP Over Ethernet	3-8
Templates	3-9
PPP Callback	3-11
Magic Number	3-12
Authentication Protocols	3-12
Password Authentication Protocol (PAP)	3-12
Challenge-Handshake Authentication Protocol (CHAP)	3-13
Configuring Authentication	3-13
Assigning IP Addresses	3-15
PPP Link Management	3-16
Debugging PPP Links	3-17
Support for PPP	3-17
Configuration Examples	3-20
Configuring a PPP link	3-20
Multilink Aggregation	3-22
Dial on Demand Links	3-24
Link Quality Monitoring	3-24
Compression	3-25
Bandwidth on Demand	3-25
Command Reference	3-27
ACTIVATE PPP	3-27
ADD PPP	3-28
CREATE PPP	3-31
CREATE PPP TEMPLATE	3-36
DELETE PPP	3-41
DESTROY PPP	3-41
DESTROY PPP TEMPLATE	3-42
DISABLE PPP	3-42
DISABLE PPP DEBUG	3-43
DISABLE PPP TEMPLATE DEBUG	3-43
ENABLE PPP	3-44
ENABLE PPP DEBUG	3-45
ENABLE PPP TEMPLATE DEBUG	3-46
PURGE PPP	3-47
RESET PPP	3-48
SET PPP	3-49
SET PPP TEMPLATE	3-54
SHOW PPP	3-58
SHOW PPP CONFIG	3-59
SHOW PPP COUNT	3-65
SHOW PPP DEBUG	3-75
SHOW PPP IDLETIMER	3-76
SHOW PPP LIMITS	3-77
SHOW PPP MULTILINK	3-78
SHOW PPP NAMESERVER	3-80
SHOW PPP TEMPLATE	3-80
SHOW PPP TXSTATUS	3-84

CHAPTER 4 Integrated Services Digital Network (ISDN)

Introduction	4-3
Basic Rate Access	4-3
Support for ISDN	4-6
BRI Physical Layer	4-7
Configuring and Controlling the Basic Rate Interface	4-7
Examining the Status of the Basic Rate Interface	4-9
Monitoring Operation of the Basic Rate Interface	4-11
LAPD	4-11
Operation	4-11
Packet mode support	4-12
Fault Finding	4-12
Default Setup	4-13
Addressing	4-13
Frame Control Fields	4-14
Non-Associated Signalling	4-15
Q.931	4-15
Service Profile Identifiers (SPIDs)	4-17
Profiles Which Require SPIDs	4-17
Definition of SPIDs	4-17
SPID Initialisation	4-18
SPID Debugging	4-19
Automatic Switch Detection	4-21
Call Control	4-22
Remote CAPI	4-26
Call Logging	4-27
Using a Domain Name Server	4-27
Slotted Interface Numbering	4-28
Always On/Dynamic ISDN (AODI)	4-28
Components of AODI	4-28
Configuring AODI	4-29
Data Over Voice	4-32
Configuration Examples	4-33
A Basic ISDN Setup	4-33
Refining the ISDN Setup	4-41
Command Reference	4-42
ACTIVATE ISDN CALL	4-42
ACTIVATE Q931 ASPID	4-43
ACTIVATE Q931 MESSAGE	4-43
ADD ISDN CALL	4-44
ADD ISDN CLILIST	4-50
ADD ISDN DOMAINNAME	4-50
ADD LAPD TEI	4-51
ADD LAPD XSPID	4-51
ADD LAPD XTEI	4-52
DEACTIVATE ISDN CALL	4-52
DELETE ISDN CALL	4-53
DELETE ISDN CLILIST	4-53
DELETE ISDN DOMAINNAME	4-54
DELETE LAPD TEI	4-54
DELETE LAPD XSPID	4-55
DELETE LAPD XTEI	4-55
DISABLE BRI CTEST	4-56
DISABLE BRI DEBUG	4-56
DISABLE BRI TEST	4-57
DISABLE ISDN CALL	4-58
DISABLE ISDN LOG	4-58
DISABLE Q931 DEBUG	4-59
DISABLE RAPI	4-60

ENABLE BRI CTEST	4-60
ENABLE BRI DEBUG	4-61
ENABLE BRI TEST	4-62
ENABLE ISDN CALL	4-65
ENABLE ISDN LOG	4-65
ENABLE Q931 ASPID	4-66
ENABLE Q931 DEBUG	4-66
ENABLE RAPI	4-71
RESET BRI	4-72
RESET BRI COUNTERS	4-72
RESET Q931	4-73
SET BRI	4-73
SET ISDN CALL	4-75
SET ISDN DOMAINNAME	4-80
SET ISDN LOG	4-81
SET LAPD	4-82
SET Q931	4-84
SHOW BRI CONFIGURATION	4-86
SHOW BRI COUNTERS	4-88
SHOW BRI CTEST	4-94
SHOW BRI DEBUG	4-95
SHOW BRI STATE	4-96
SHOW BRI TEST	4-100
SHOW ISDN CALL	4-103
SHOW ISDN CLILIST	4-107
SHOW ISDN DOMAINNAME	4-108
SHOW ISDN LOG	4-109
SHOW LAPD	4-110
SHOW LAPD COUNT	4-112
SHOW LAPD STATE	4-114
SHOW Q931	4-114
SHOW Q931 SPID	4-117

CHAPTER 5 **X.25**

Introduction	5-2
DTE Mode	5-3
DTE Addresses	5-4
Encapsulations	5-4
Configuring X.25 DTE	5-6
Configure the X.25 DTE Interface	5-6
Configuring Call Parameter Entries	5-7
Configuring Permanent Virtual Circuits	5-7
Configuration Examples	5-7
Command Reference	5-8
ACTIVATE MIOX CIRCUIT	5-9
ADD MIOX CIRCUIT	5-10
ADD X25T CPAR	5-11
CREATE X25T	5-12
DEACTIVATE MIOX CIRCUIT	5-14
DELETE MIOX CIRCUIT	5-15
DELETE X25T CPAR	5-15
DESTROY X25T	5-16
DISABLE MIOX CIRCUIT	5-16
ENABLE MIOX CIRCUIT	5-17
RESET X25T	5-18
SET MIOX	5-18
SET MIOX CIRCUIT	5-19
SET X25T	5-21

SET X25T CPAR	5-22
SHOW MIOX	5-23
SHOW MIOX COUNT	5-24
SHOW MIOX CIRCUIT	5-26
SHOW X25T	5-30
SHOW X25T CPAR	5-34

CHAPTER 6 Internet Protocol (IP)

Introduction	6-4
The Internet	6-4
Addressing	6-6
Subnets	6-8
Multihoming	6-9
Address Resolution	6-9
DHCP Client	6-11
ICMP	6-11
Routing	6-12
Routing Information Filters	6-13
RIP	6-14
Metrics	6-16
Policy-Based Routing	6-16
Priority-Based Routing	6-18
Route Templates	6-19
Named Hosts	6-20
DNS Relay Agent	6-21
Traffic Filters	6-21
SNMP	6-23
Control and Debug Commands	6-24
Ping and Trace Route	6-25
Security Options	6-26
Broadcast Forwarding	6-26
Examples	6-27
BOOTP Relay Agent	6-29
IP Multicasting	6-31
Remote Address Assignment	6-32
IP Address Pools	6-32
Configuration Examples	6-33
A Basic TCP/IP Setup	6-33
Troubleshooting	6-37
Configuring IP Filters	6-38
Command Reference	6-43
ADD BOOTP RELAY	6-43
ADD IP ARP	6-44
ADD IP FILTER	6-45
ADD IP HELPER	6-51
ADD IP HOST	6-52
ADD IP INTERFACE	6-53
ADD IP RIP	6-56
ADD IP ROUTE	6-57
ADD IP ROUTE FILTER	6-59
ADD IP ROUTE TEMPLATE	6-60
ADD IP TRUSTED	6-61
CREATE IP POOL	6-62
DELETE BOOTP RELAY	6-63
DELETE IP ARP	6-63
DELETE IP FILTER	6-64
DELETE IP HELPER	6-64
DELETE IP HOST	6-65

DELETE IP INTERFACE	6-66
DELETE IP RIP	6-67
DELETE IP ROUTE	6-68
DELETE IP ROUTE FILTER	6-69
DELETE IP ROUTE TEMPLATE	6-69
DELETE IP TRUSTED	6-69
DELETE TCP	6-70
DESTROY IP POOL	6-71
DISABLE BOOTP RELAY	6-71
DISABLE IP	6-72
DISABLE IP DEBUG	6-72
DISABLE IP DNSRELAY	6-73
DISABLE IP ECHOREPLY	6-73
DISABLE IP FOFILTER	6-73
DISABLE IP FORWARDING	6-74
DISABLE IP HELPER	6-74
DISABLE IP INTERFACE	6-75
DISABLE IP REMOTEASSIGN	6-75
DISABLE IP ROUTE	6-76
DISABLE IP SRCROUTE	6-76
ENABLE BOOTP RELAY	6-77
ENABLE IP	6-77
ENABLE IP DEBUG	6-78
ENABLE IP DNSRELAY	6-78
ENABLE IP ECHOREPLY	6-78
ENABLE IP FOFILTER	6-78
ENABLE IP FORWARDING	6-79
ENABLE IP HELPER	6-80
ENABLE IP INTERFACE	6-80
ENABLE IP REMOTEASSIGN	6-81
ENABLE IP ROUTE	6-81
ENABLE IP SRCROUTE	6-82
PING	6-82
PURGE BOOTP RELAY	6-84
PURGE IP	6-84
RESET IP	6-84
RESET IP COUNTER	6-85
RESET IP INTERFACE	6-85
SET BOOTP MAXHOPS	6-86
SET IP ARP	6-86
SET IP AUTONOMOUS	6-87
SET IP FILTER	6-88
SET IP HOST	6-91
SET IP INTERFACE	6-92
SET IP LOCAL	6-94
SET IP NAMESERVER	6-95
SET IP RIP	6-96
SET IP RIPTIMER	6-97
SET IP ROUTE	6-98
SET IP ROUTE FILTER	6-100
SET IP ROUTE TEMPLATE	6-101
SET IP SECONDARYNAMESERVER	6-102
SET PING	6-103
SET TRACE	6-104
SHOW BOOTP RELAY	6-105
SHOW IP	6-106
SHOW IP ARP	6-108
SHOW IP COUNTER	6-109
SHOW IP DEBUG	6-116

SHOW IP FILTER	6-117
SHOW IP HELPER	6-119
SHOW IP HOST	6-120
SHOW IP INTERFACE	6-121
SHOW IP POOL	6-124
SHOW IP RIP	6-126
SHOW IP RIPTIMER	6-127
SHOW IP RIP COUNTER	6-128
SHOW IP ROUTE	6-130
SHOW IP ROUTE FILTER	6-134
SHOW IP ROUTE TEMPLATE	6-135
SHOW IP TRUSTED	6-136
SHOW IP UDP	6-136
SHOW PING	6-137
SHOW TCP	6-139
SHOW TRACE	6-143
STOP PING	6-145
STOP TRACE	6-145
TRACE	6-146

CHAPTER 7 **Terminal Server**

Introduction	7-2
TTY Devices	7-2
Command Line Editing and Recall	7-4
Accessing Telnet Hosts	7-5
Command Reference	7-6
SET TELNET	7-6
SET TTY	7-7
SHOW TTY	7-8
TELNET	7-11

CHAPTER 8 **Compression Services**

Introduction	8-2
Data Compression	8-2
ENCO Services	8-4
Compression	8-4
User Modules	8-5
PPP	8-5
X.25 Link Compression	8-5
Command Reference	8-5
DISABLE ENCO COMPSTATISTICS	8-5
DISABLE ENCO DEBUGGING	8-6
ENABLE ENCO COMPSTATISTICS	8-6
ENABLE ENCO DEBUGGING	8-7
RESET ENCO COUNTERS	8-7
SET ENCO SW	8-7
SHOW ENCO	8-8
SHOW ENCO CHANNEL	8-9
SHOW ENCO COUNTERS	8-13

CHAPTER 9 **Test Facility**

Introduction	9-2
Ethernet Port Tests	9-4
Asynchronous Port Tests	9-6
Basic Rate ISDN Port Tests	9-7
Command Reference	9-8
DISABLE TEST INTERFACE	9-8

	ENABLE TEST INTERFACE	9-9
	RESET TEST INTERFACE	9-10
	SHOW TEST	9-10
CHAPTER 10	Trigger Facility	
	Introduction	10-2
	Defining Triggers	10-3
	Examples	10-3
	Initiating ISDN Calls During Off-Peak Periods	10-3
	Retrieving System Snapshots After a Reboot	10-5
	Command Reference	10-5
	ACTIVATE TRIGGER	10-5
	ADD TRIGGER	10-6
	CREATE TRIGGER	10-7
	DELETE TRIGGER	10-11
	DESTROY TRIGGER	10-11
	DISABLE TRIGGER	10-12
	ENABLE TRIGGER	10-12
	PURGE TRIGGER	10-13
	SET TRIGGER	10-13
	SHOW TRIGGER	10-16
CHAPTER 11	Time Division Multiplexing (TDM)	
	Introduction	11-2
	Configuration Examples	11-2
	Command Reference	11-3
	ADD TDM	11-3
	CREATE TDM	11-4
	DELETE TDM	11-5
	DESTROY TDM	11-5
	PURGE TDM	11-6
	SHOW TDM	11-6
CHAPTER 12	Logging Facility	
	Introduction	12-2
	Format of Log Messages	12-3
	Secure Router Log Protocol	12-4
	Net Manage Message Protocol	12-5
	Processing of Log Messages	12-5
	Output Definitions and Message Filters	12-5
	Destinations	12-6
	Configuring Output Definitions	12-8
	Configuring Message Filters	12-8
	Configuration Example	12-9
	Command Reference	12-12
	ADD LOG OUTPUT	12-12
	ADD LOG RECEIVE	12-15
	CREATE LOG OUTPUT	12-16
	DELETE LOG OUTPUT	12-19
	DELETE LOG RECEIVE	12-20
	DESTROY LOG OUTPUT	12-20
	DISABLE LOG	12-21
	DISABLE LOG GENERATION	12-21
	DISABLE LOG OUTPUT	12-21
	DISABLE LOG RECEPTION	12-22
	ENABLE LOG	12-22
	ENABLE LOG GENERATION	12-23

ENABLE LOG OUTPUT	12-23
ENABLE LOG RECEPTION	12-23
FLUSH LOG OUTPUT	12-24
PURGE LOG	12-24
SET LOG OUTPUT	12-25
SET LOG RECEIVE	12-29
SET LOG UTCOFFSET	12-30
SHOW LOG	12-31
SHOW LOG COUNTERS	12-38
SHOW LOG OUTPUT	12-40
SHOW LOG QUEUE	12-44
SHOW LOG RECEIVE	12-45
SHOW LOG STATUS	12-46
 CHAPTER 13 Scripting	
Introduction	13-2
Creating Scripts	13-2
Script Commands	13-2
Using the Built-in Text Editor	13-3
Loading from a TFTP Server	13-3
Loading from an Asynchronous Port	13-3
Using Scripts	13-4
Script Parameters	13-4
Script Control Structures	13-4
Command Reference	13-5
ACTIVATE SCRIPT	13-5
ADD SCRIPT	13-6
DEACTIVATE SCRIPT	13-7
DELETE SCRIPT	13-8
IF..THEN..ELSE..ENDIF	13-8
SET SCRIPT	13-10
SHOW SCRIPT	13-11
WAIT	13-12
 CHAPTER 14 Telephony Services	
Introduction	14-2
Ports	14-2
Extensions	14-3
Groups	14-4
Numbers	14-4
Tones	14-4
Calls	14-6
Call Handling	14-7
Call Waiting	14-8
Conference Calling	14-8
Call Transfer	14-9
Call Forwarding	14-9
Call Processing	14-9
MSN and DDI Support	14-9
B Channel Allocation	14-10
Bearer Capability, LLC and HLC	14-11
Enbloc or Overlap Dialling	14-12
Tone Suppression	14-12
Hardware Configuration	14-12
Call Logging	14-12
Command Reference	14-13
CREATE PBX EXTENSION	14-13
CREATE PBX GROUP	14-16

DESTROY PBX EXTENSION	14-17
DESTROY PBX GROUP	14-18
DISABLE PBX DEBUG	14-18
ENABLE PBX DEBUG	14-19
SET PBX	14-20
SET PBX EXTENSION	14-21
SET PBX GROUP	14-24
SHOW PBX	14-25
SHOW PBX CALL	14-26
SHOW PBX EXTENSION	14-27
SHOW PBX GROUP	14-29

CHAPTER 15 **Dynamic Host Configuration Protocol (DHCP)**

Introduction	15-2
The Dynamic Host Configuration Protocol (DHCP)	15-2
Configuration Example	15-3
Command Reference	15-4
ADD DHCP POLICY	15-4
ADD DHCP RANGE	15-9
CREATE DHCP POLICY	15-9
CREATE DHCP RANGE	15-10
DELETE DHCP POLICY	15-11
DELETE DHCP RANGE	15-15
DESTROY DHCP POLICY	15-15
DESTROY DHCP RANGE	15-16
DISABLE DHCP	15-16
ENABLE DHCP	15-17
SET DHCP POLICY	15-17
SHOW DHCP	15-22
SHOW DHCP CLIENT	15-23
SHOW DHCP POLICY	15-24
SHOW DHCP RANGE	15-25

CHAPTER 16 **Simple Network Management Protocol (SNMP)**

Introduction	16-2
Network Management Framework	16-2
Structure of Management Information	16-3
Names	16-4
Instances	16-4
Syntax	16-5
Access	16-5
Status	16-5
Description	16-6
The SNMP Protocol	16-6
SNMP Messages	16-6
Polling versus Event Notification	16-8
Communities and Views	16-8
Support for SNMP	16-9
Configuration Example	16-11
Command Reference	16-12
ADD SNMP COMMUNITY	16-13
CREATE SNMP COMMUNITY	16-14
DELETE SNMP COMMUNITY	16-15
DESTROY SNMP COMMUNITY	16-16
DISABLE SNMP	16-16
DISABLE SNMP AUTHENTICATE_TRAP	16-16
DISABLE SNMP COMMUNITY	16-17
ENABLE SNMP	16-17

	ENABLE SNMP AUTHENTICATE_TRAP	16-18
	ENABLE SNMP COMMUNITY	16-18
	SET SNMP COMMUNITY	16-19
	SHOW SNMP	16-20
	SHOW SNMP COMMUNITY	16-22
CHAPTER 17	Firewall	
	Introduction	17-2
	Policies	17-3
	Rules	17-4
	NAT	17-6
	Monitoring Firewall Activity	17-6
	Debugging	17-6
	Logging	17-6
	Configuration Examples	17-8
	Minimum Configuration for a Small Office	17-8
	A Firewall with an ISP-assigned Internet Address	17-9
	A Firewall with a Single Global Internet Address	17-10
	Allowing Access to a WWW Server	17-10
	Command Reference	17-11
	ADD FIREWALL POLICY INTERFACE	17-11
	ADD FIREWALL POLICY NAT	17-12
	ADD FIREWALL POLICY RULE	17-14
	CREATE FIREWALL POLICY	17-16
	DELETE FIREWALL POLICY INTERFACE	17-17
	DELETE FIREWALL POLICY NAT	17-18
	DELETE FIREWALL POLICY RULE	17-19
	DELETE FIREWALL SESSION	17-19
	DESTROY FIREWALL POLICY	17-20
	DISABLE FIREWALL	17-20
	DISABLE FIREWALL POLICY	17-21
	ENABLE FIREWALL	17-22
	ENABLE FIREWALL POLICY	17-22
	SET FIREWALL POLICY RULE	17-24
	SHOW FIREWALL	17-25
	SHOW FIREWALL POLICY	17-26
	SHOW FIREWALL SESSION	17-32
CHAPTER 18	Link Compression	
	Introduction	18-2
	Overview	18-2
	Link Compression	18-2
	PPP	18-3
	X.25	18-4
APPENDIX A	Messages	
	Introduction	A-2
	Message Descriptions	A-2
	smm001–smm255: Global Messages	A-2
	s03256–s03999: Point-to-Point Protocol	A-7
	s05256–s05999: Internet Protocol (IP)	A-10
	s14256–s14999: Q.931	A-15
	s18256–s18999: TEST Module	A-16
	s19256–s19999: LAPD	A-18
	s22256–s22999: TCP	A-20
	s23256–s23999: Ethernet Driver	A-20
	s28256–s28999: Compression	A-21

s30256–s30999: X.25 Layer 3 (DTE)	A-22
s31256–s31999: FLASH Driver	A-24
s33256–s33999: TELNET	A-25
s34256–s34999: System	A-25
s35256–s35999: Command Processor	A-26
s36256–s36999: TTY	A-26
s37256–s37999: ISDN Call Control	A-27
s38256–s38999: MIOX	A-30
s39256–s39999: BOOTP	A-31
s41256–s41999: BRI Driver	A-32
s43256–s43999: PORT Driver	A-33
s45256–s45999: User Authentication Facility	A-35
s48256–s48999: LOADER	A-38
s49256–s49999: INSTALL	A-40
s53256–s53999: Trigger Facility	A-41
s54256–s54999: Scripting	A-43
s55256–s55999: Time Division Multiplexing (TDM)	A-43
s56256–s56999: File Subsystem	A-45
s57256–s57999: Logging Facility	A-46
s58256–s58999: PING	A-49
s59256–s59999: Simple Network Management Protocol (SNMP)	A-50
s61256–s61999: Telephony Services	A-51
s70256–s70999: Dynamic Host Configuration Protocol (DHCP)	A-52
s77256–s77999: Firewall	A-53

APPENDIX B Reference Tables

Module Identifiers and Names	B-2
FLASH File System Message Codes	B-4
ISDN Q.931 Call Clearance Cause Codes	B-6
Log Message Types and Subtypes	B-8

APPENDIX C SNMP MIBs

Introduction	C-2
Allied Telesyn Enterprise MIB	C-3
The Products Sub-tree	C-3
The AT Router Sub-tree	C-4
The Objects Group	C-4
The arlInterfaces Group	C-6
The Modules Group	C-7
MIB-II MIB	C-11
Implementation	C-12
Ethernet-like Interface Types MIB	C-13
Implementation	C-13
Host Resources MIB	C-14
Implementation	C-15

Glossary

Index

List of Figures

1-1	Using the question mark character (“?”) to display help for the current command.	1-6
1-2	A typical login session for user BRUCE on router CMD.	1-7
1-3	The editor screen layout.	1-18
1-4	Logging in to the router from a web browser.	1-19
1-5	Router startup messages.	1-21
1-6	Example output from the DUMP command.	1-36
1-7	The editor screen layout.	1-39
1-8	Example output from the SHOW ALIAS command.	1-59
1-9	Example output from the SHOW BUFFER command.	1-60
1-10	Example output from the SHOW BUFFER SCAN command.	1-60
1-11	Example output from the SHOW BUFFER SCAN command for a specified address.	1-61
1-12	Example output from the SHOW BUFFER SCAN QUEUEPOINTERS command.	1-61
1-13	Example output from the SHOW CONFIG command.	1-62
1-14	Example output from the SHOW CPU command.	1-63
1-15	Example output from the SHOW EXCEPTION command.	1-65
1-16	Example output from the SHOW FFIL command.	1-66
1-17	Example output from the SHOW FILE command.	1-67
1-18	Example output from the SHOW FLASH command.	1-68
1-19	Example output from the SHOW FLASH PHYSICAL command.	1-69
1-20	Example output from the SHOW HTTP CLIENT command.	1-70
1-21	Example output from the SHOW HTTP DEBUG command.	1-70
1-22	Example output from the SHOW HTTP SESSION command.	1-71
1-23	Example output from the SHOW HTTP SERVER command.	1-72
1-24	Example output from the SHOW INSTALL command.	1-75
1-25	Example output from the SHOW LOADER command.	1-76
1-26	Example output from the SHOW STARTUP command.	1-78
1-27	Example output from the SHOW SYSTEM command.	1-78
1-28	Example output from the SHOW USER command.	1-80
1-29	Example output from the SHOW USER CONFIGURATION command.	1-81
2-1	Format of an Ethernet frame.	2-4
2-2	Format of an Ethernet frame with SNAP encapsulation.	2-5
2-3	Example output from the SHOW ETH CONFIGURATION command.	2-22
2-4	Example output from the SHOW ETH COUNTERS=COLLISIONS command.	2-23
2-5	Example output from the SHOW ETH COUNTERS=DIAGNOSTIC command for 68360-based hardware.	2-24
2-6	Example output from the SHOW ETH COUNTERS=DIAGNOSTIC command for SONIC-based hardware.	2-24
2-7	Example output from the SHOW ETH COUNTERS=DOT3STAT command.	2-26
2-8	Example output from the SHOW ETH COUNTERS=INTERFACE command.	2-27
2-9	Example output from the SHOW ETH MACADDRESS command.	2-29

2-10	Example output from the SHOW ETH RECEIVE command.....	2-30
2-11	Example output from the SHOW INTERFACE command.....	2-30
2-12	Example output from the SHOW INTERFACE command for a specific interface.....	2-31
2-13	Example output from the SHOW INTERFACE COUNTERS command.	2-32
2-14	Example output from the SHOW PORT command.	2-34
2-15	Example output from the SHOW PORT COUNTERS command.	2-37
2-16	Example output from the SHOW PORT HISTORY command.....	2-38
2-17	Example output from the SHOW PORT SUMMARY command.	2-38
3-1	The Password Authentication Protocol (PAP) authentication process.....	3-12
3-2	The Challenge Handshake Authentication Protocol (CHAP) authentication process.....	3-13
3-3	Example output from the SHOW PPP command for a PPP link.	3-21
3-4	Example output from the SHOW IP INTERFACE command for a PPP link configured for use by the IP routing module.	3-22
3-5	Example output from a SHOW PPP command for a PPP interface aggregated over two ISDN B channels.....	3-23
3-6	Example output from the SHOW ISDN CALL command for a PPP interface aggregated over two ISDN B channels.....	3-23
3-7	Example configuration for bandwidth on demand.....	3-25
3-8	Example output from the SHOW PPP command.	3-59
3-9	Example output from the SHOW PPP CONFIG command.....	3-60
3-10	Example output from the SHOW PPP COUNT=INTERFACE command.	3-65
3-11	Example output from the SHOW PPP COUNT=LCP command.....	3-67
3-12	Example output from the SHOW PPP COUNT=MULTILINK command.....	3-71
3-13	Example output from the SHOW PPP COUNT=NCP command.....	3-72
3-14	Example output from the SHOW PPP DEBUG command.....	3-75
3-15	Example output from the SHOW PPP IDLETIMER command.....	3-76
3-16	Example output from the SHOW PPP Limits command.	3-77
3-17	Example output from the SHOW PPP MULTILINK command.	3-78
3-18	Example output from the SHOW PPP NAMESERVER command.....	3-80
3-19	Example output from the SHOW PPP TEMPLATE command.....	3-81
3-20	Example output from the SHOW PPP TEMPLATE DEBUG command.....	3-83
3-21	Example output from the SHOW PPP TXSTATUS command.	3-84
4-1	A typical ISDN Basic Rate Access circuit.	4-4
4-2	Example configuration for a basic ISDN network.	4-33
4-3	Example output from the SHOW ISDN CALL command for Head Office.....	4-37
4-4	Example commands and output to test the configuration of the central site router in a basic ISDN network.....	4-39
4-5	Example commands and output to test the configuration of the regional site router in a basic ISDN network.....	4-40
4-6	Example output from the ENABLE Q931 DEBUG=MDECODE command for a call initiated by the router.....	4-67
4-7	Example output from the ENABLE Q931 DEBUG=MRAW command for a call initiated by the router.....	4-68
4-8	Example output from the ENABLE Q931 DEBUG=SDLC command.....	4-69
4-9	Example output from the ENABLE Q931 DEBUG=SSPID command.	4-69
4-10	Example output from the ENABLE Q931 DEBUG=SSPIDFILE command.	4-70
4-11	Example output from the ENABLE Q931 DEBUG=STATE command.	4-70
4-12	Example output from the SHOW BRI CONFIGURATION command.....	4-87
4-13	Example output from the SHOW BRI COUNTERS=INTERFACE command.	4-88
4-14	Example output from the SHOW BRI COUNTERS=BRI command for an S/T interface.	4-90
4-15	Example output from the SHOW BRI COUNTERS=BRI command for a U interface.....	4-91
4-16	Example output from the SHOW BRI CTEST command.	4-94
4-17	Example output from the SHOW BRI DEBUG command.....	4-95
4-18	Example output from the SHOW BRI STATE command for an S/T interface.....	4-96
4-19	Example output from the SHOW BRI STATE command for a U interface.	4-98
4-20	Example output from the SHOW BRI TEST command for BRI interfaces using an MC145474 controller.	4-100

4-21	Example output from the SHOW BRI TEST command for BRI interfaces using a PSB2186 controller.	4-101
4-22	Example output from the SHOW BRI TEST command for BRI interfaces using a PEB2091 controller.	4-102
4-23	Example output from the SHOW BRI TEST command for BRI interfaces using a MC145572 controller.	4-102
4-24	Example output from the SHOW ISDN CALL command.	4-104
4-25	Example output from the SHOW ISDN CALL command for a specified call name.	4-105
4-26	Example output from the SHOW ISDN CLILIST command.	4-108
4-27	Example output from the SHOW ISDN DOMAINNAME command.	4-108
4-28	Example output from the SHOW ISDN LOG command.	4-109
4-29	Example output from the SHOW LAPD command for a Basic Rate Interface.	4-110
4-30	Example output from the SHOW LAPD command for a Primary Rate Interface.	4-111
4-31	Example output from the SHOW LAPD COUNT command.	4-113
4-32	Example output from the SHOW LAPD STATE command.	4-114
4-33	Example output from the SHOW Q931 command.	4-115
4-34	Example output from the SHOW Q931 CALL command.	4-117
4-35	Example output from the SHOW Q931 SPID command.	4-118
4-36	Example output from the SHOW Q931 SPID command during the auto-SPID procedure.	4-119
5-1	Example output from the SHOW MIOX command.	5-23
5-2	Example output from the SHOW MIOX COUNT command.	5-24
5-3	Example output from the SHOW MIOX CIRCUIT command.	5-27
5-4	Example output from the SHOW MIOX CIRCUIT COUNTER command.	5-27
5-5	Example output from the SHOW MIOX CIRCUIT ENCAP command.	5-29
5-6	Example output from the SHOW X25T command.	5-30
5-7	Example output from the SHOW X25T CIRCUIT command.	5-32
5-8	Example output from the SHOW X25T COUNT command.	5-33
5-9	Example output from the SHOW X25T CPAR command.	5-34
6-1	Format of an IP datagram.	6-5
6-2	Subdivision of the 32 bits of an Internet address into network and host fields for class A, B and C networks.	6-6
6-3	Example configuration for broadcast forwarding to a unicast address.	6-27
6-4	Example configuration for broadcast forwarding to a multicast address.	6-28
6-5	Example configuration for a basic TCP/IP network.	6-33
6-6	Example output from the SHOW IP ROUTE command for a basic TCP/IP network.	6-36
6-7	Example output from the SHOW PPP command for a basic TCP/IP network.	6-37
6-8	Example output from the SHOW IP RIP command for a basic TCP/IP network.	6-37
6-9	Example output from the PING command.	6-38
6-10	Example configuration for IP filtering.	6-39
6-11	Example output from the SHOW IP FILTER command for IP filtering.	6-42
6-12	Example output from the SHOW IP INTERFACE command for IP filtering.	6-42
6-13	Example output from the PING command when SCREENOUTPUT is set to YES.	6-83
6-14	Example output from the SHOW BOOTP RELAY command.	6-105
6-15	Example output from the SHOW IP command.	6-107
6-16	Example output from the SHOW IP ARP command.	6-108
6-17	Example output from the SHOW IP COUNTER=ICMP command.	6-110
6-18	Example output from the SHOW IP COUNTER=INTERFACE command.	6-111
6-19	Example output from the SHOW IP COUNTER=IP command.	6-112
6-20	Example output from the SHOW IP COUNTER=MULTICAST command.	6-113
6-21	Example output from the SHOW IP COUNTER=RIP command.	6-114
6-22	Example output from the SHOW IP COUNTER=ROUTE command.	6-115
6-23	Example output from the SHOW IP COUNTER=UDP command.	6-115
6-24	Example output from the SHOW IP FILTER command.	6-117
6-25	Example output from the SHOW IP HELPER command.	6-119
6-26	Example output from the SHOW IP HELPER COUNTER command.	6-119
6-27	Example output from the SHOW IP HOST command.	6-120
6-28	Example output from the SHOW IP INTERFACE command.	6-121

6-29	Example output from the SHOW IP INTERFACE COUNTER command.....	6-123
6-30	Example output from the SHOW IP POOL command.	6-125
6-31	Example output from the SHOW IP POOL SUMMARY command.	6-125
6-32	Example output from the SHOW IP RIP command.	6-126
6-33	Example output from the SHOW IP RIPTIMER command.....	6-127
6-34	Example output from the SHOW IP RIP COUNTER=DETAIL command.	6-129
6-35	Example output from the SHOW IP ROUTE command.	6-131
6-36	Example output from the SHOW IP ROUTE GENERAL command.....	6-132
6-37	Example output from the SHOW IP ROUTE CACHE command.	6-132
6-38	Example output from the SHOW IP ROUTE COUNT command.....	6-133
6-39	Example output from the SHOW IP ROUTE FILTER command.....	6-134
6-40	Example output from the SHOW IP ROUTE TEMPLATE command.	6-135
6-41	Example output from the SHOW IP ROUTE TEMPLATE command.	6-135
6-42	Example output from the SHOW IP TRUSTED command.	6-136
6-43	Example output from the SHOW IP UDP command.	6-136
6-44	Example output from the SHOW PING command.	6-138
6-45	Example output from the SHOW TCP command for a specified TCP connection.	6-140
6-46	Example output from the SHOW TCP command.....	6-141
6-47	Example output from the SHOW TRACE command.	6-144
7-1	TTY devices provide an interface between terminals and Telnet connections, the router's command processor, and interactive and Telnet services provided by the router.	7-2
7-2	Example output from the SHOW TTY command.....	7-8
7-3	Example output from the SHOW TTY=ALL SUMMARY command.....	7-9
7-4	Example output from the SHOW TTY DEFAULT command.....	7-10
7-5	Example output from the TELNET command.....	7-11
8-1	Example output from the SHOW ENCO command.	8-8
8-2	Example output from the SHOW ENCO CHANNEL command.....	8-9
8-3	Example output from the SHOW ENCO CHANNEL command for a specified channel.	8-10
8-4	Example output from the SHOW ENCO CHANNEL COUNTERS command.....	8-11
8-5	Example output from the SHOW ENCO COUNTERS=JOBPROCESSING command.....	8-14
8-6	Example output from the SHOW ENCO COUNTERS=STAC command.....	8-16
8-7	Example output from the SHOW ENCO COUNTERS=USER command.	8-19
8-8	Example output from the SHOW ENCO COUNTERS=UTIL command.	8-20
9-1	Pin wiring diagram for a cable to connect a terminal to an asynchronous port that is to be tested by the Test Facility.	9-3
9-2	Pin wiring diagram for a cable to connect a terminal to a DB9 female asynchronous port that is to be tested by the Test Facility.	9-4
9-3	Pin wiring diagram for a cable to connect a terminal to a DB9 male asynchronous port that is to be tested by the Test Facility.	9-4
9-4	Ethernet AUI loopback plug wiring diagram.	9-5
9-5	Ethernet twisted pair (TP) loopback plug wiring diagram.	9-5
9-6	RJ45 loopback plug wiring diagram.	9-6
9-7	DB9 male loopback plug wiring diagram.	9-6
9-8	DB9 female loopback plug wiring diagram.	9-6
9-9	Basic Rate ISDN loopback plug wiring diagram.	9-7
9-10	Example output from the ENABLE TEST INTERFACE MORE command for an asynchronous port. .	9-10
9-11	Example output from the SHOW TEST INTERFACE command.	9-11
9-12	Example output from the SHOW TEST INTERFACE COUNTERS command.	9-12
10-1	Triggers respond to events by performing a sequence of predefined scripts.....	10-2
10-2	The effects of different combinations of the AFTER and BEFORE parameters in the CREATE TRIGGER and SET TRIGGER commands.	10-9
10-3	Example output from the SHOW TRIGGER command.....	10-17
10-4	Example output from the SHOW TRIGGER FULL command.....	10-17
10-5	Example output from the SHOW TRIGGER STATUS command.	10-19
10-6	Example output from the SHOW TRIGGER COUNTER command.....	10-20

11-1	Example output from the SHOW TDM GROUP command.....	11-7
12-1	Example configuration for a basic logging facility.	12-10
12-2	Example output from the SHOW LOG command.....	12-34
12-3	Example output from the SHOW LOG FULL command.....	12-36
12-4	Example output from the SHOW LOG COUNTERS command.....	12-38
12-5	Example output from the SHOW LOG OUTPUT command.	12-40
12-6	Example output from the SHOW LOG OUTPUT FULL command.....	12-42
12-7	Example output from the SHOW LOG QUEUE command.....	12-44
12-8	Example output from the SHOW LOG RECEIVE command.	12-45
12-9	Example output from the SHOW LOG STATUS command.....	12-46
13-1	Example output from the SHOW SCRIPT command.....	13-11
13-2	Example output from the SHOW SCRIPT command for a specified script.	13-12
14-1	Bell Tone.....	14-5
14-2	Unavailable Tone.	14-5
14-3	External Dial Tone.	14-5
14-4	Example output from the SHOW PBX command.....	14-25
14-5	Example output from the SHOW PBX CALL command.....	14-26
14-6	Example output from the SHOW PBX EXTENSION command.	14-27
14-7	Example output from the SHOW PBX GROUP command.	14-29
15-1	Example output from the SHOW DHCP command.....	15-22
15-2	Example output from the SHOW DHCP CLIENT command.....	15-23
15-3	Example output from the SHOW DHCP POLICY command.	15-24
15-4	Example output from the SHOW DHCP RANGE command.	15-25
16-1	Components of a network management system.	16-2
16-2	The top levels of the Internet-standard Management Information Base (MIB).	16-3
16-3	Format of an SNMP message.	16-7
16-4	Example output from the SHOW SNMP command.	16-20
16-5	Example output from the SHOW SNMP COMMUNITY command.	16-22
17-1	Example output from the SHOW FIREWALL command.	17-25
17-2	Example output from the SHOW FIREWALL POLICY command.....	17-27
17-3	Example output from the SHOW FIREWALL POLICY COUNTERS command.....	17-29
17-4	Example output from the SHOW FIREWALL SESSION command.	17-32
C-1	The Allied Telesyn Enterprise MIB sub-tree of the Internet-standard Management Information Base (MIB).	C-3
C-2	The arInterfaces Group object tree for an AR720 with an Ethernet PIC in Expansion Bay 1.	C-7
C-3	The MIB-II sub-tree of the Internet-standard Management Information Base (MIB).	C-11
C-4	The Ethernet-like interface types sub-tree of the Internet-standard Management Information Base (MIB).	C-13
C-5	The Host Resources sub-tree of the Internet-standard Management Information Base (MIB).	C-14

List of Tables

I	Protocols and standards supported by the AR router.	xxxii
II	Typographic conventions used in this manual.	xxxv
1-1	Command line editing functions and keystrokes.	1-5
1-2	Secure commands controlled by the security timer.	1-10
1-3	File extensions and file types.	1-15
1-4	Router startup sequence keystrokes.	1-22
1-5	Router CPU address spaces.	1-36
1-6	Editor functions and keystrokes.	1-38
1-7	Parameters displayed in the output of the SHOW ALIAS command.	1-59
1-8	Parameters displayed in the output of the SHOW BUFFER command.	1-60
1-9	Parameters displayed in the output of the SHOW CONFIG command.	1-62
1-10	Parameters displayed in the output of the SHOW CPU command.	1-63
1-11	Parameters displayed in the output of the SHOW FFILE command.	1-66
1-12	Parameters displayed in the output of the SHOW FILE command.	1-67
1-13	Parameters displayed in the output of the SHOW FLASH command.	1-68
1-14	Parameters displayed in the output of the SHOW FLASH PHYSICAL command.	1-69
1-15	Parameters displayed in the output of the SHOW HTTP CLIENT command.	1-70
1-16	Parameters displayed in the output of the SHOW HTTP DEBUG command.	1-71
1-17	Parameters displayed in the output of the SHOW HTTP SESSION command.	1-72
1-18	Parameters displayed in the output of the SHOW HTTP SERVER command.	1-74
1-19	Parameters displayed in the output of the SHOW INSTALL command.	1-75
1-20	Parameters displayed in the output of the SHOW LOADER command.	1-76
1-21	Parameters displayed in the output of the SHOW SYSTEM command.	1-79
1-22	Parameters displayed in the output of the SHOW USER command.	1-81
1-23	Parameters displayed in the output of the SHOW USER CONFIGURATION command.	1-81
2-1	Router interface names and types.	2-2
2-2	Examples of valid interface names.	2-3
2-3	Supported Ethernet encapsulations and discriminators.	2-5
2-4	Categories of counters maintained for Ethernet interfaces.	2-6
2-5	Configurable parameters for asynchronous port.	2-8
2-6	Factory defaults for configurable parameters for asynchronous ports.	2-9
2-7	Categories of counters maintained for asynchronous ports.	2-10
2-8	Parameters displayed in the output of the SHOW ETH CONFIGURATION command.	2-22
2-9	Parameters displayed in the output of the SHOW ETH COUNTERS=DIAGNOSTIC command.	2-24
2-10	Parameters displayed in the output of the SHOW ETH COUNTERS=DOT3STAT command.	2-27
2-11	Parameters displayed in the output of the SHOW ETH COUNTERS=INTERFACE command.	2-28
2-12	Parameters displayed in the output of the SHOW INTERFACE command.	2-30
2-13	Parameters displayed in the output of the SHOW INTERFACE command for a specific interface. .	2-31
2-14	Parameters displayed in the output of the SHOW INTERFACE COUNTERS command.	2-32
2-15	Parameters displayed in the output of the SHOW PORT command.	2-35

2-16	Parameters displayed in the output of the SHOW PORT COUNTERS command.	2-37
2-17	Parameters displayed in the output of the SHOW PORT SUMMARY command.	2-38
3-1	Supported Network protocols and Network Control Protocols for the Point-to-Point Protocol.	3-4
3-2	States for control protocols of the Point-to-Point Protocol.	3-5
3-3	Example configuration parameters for bandwidth on demand.	3-26
3-4	Point-to-Point Protocol (PPP) debugging options.	3-45
3-5	Parameters displayed in the output of the SHOW PPP command.	3-59
3-6	Parameters displayed in the output of the SHOW PPP CONFIG command.	3-61
3-7	Parameters displayed in the output of the SHOW PPP COUNT=INTERFACE command.	3-66
3-8	Parameters displayed in the output of the SHOW PPP COUNT=LCP command.	3-68
3-9	Parameters displayed in the output of the SHOW PPP COUNT=MULTILINK command.	3-71
3-10	Parameters displayed in the output of the SHOW PPP COUNT=NCP command.	3-72
3-11	Parameters displayed in the output of the SHOW PPP COUNT command for BAP and BACP.	3-73
3-12	Parameters displayed in the output of the SHOW PPP DEBUG command.	3-76
3-13	Parameters displayed in the output of the SHOW PPP IDLETIMER command.	3-76
3-14	Parameters displayed the SHOW PPP LIMITS command output	3-77
3-15	Parameters displayed in the output of the SHOW PPP MULTILINK command.	3-79
3-16	Parameters displayed in the output of the SHOW PPP NAMESERVER command.	3-80
3-17	Parameters displayed in the output of the SHOW PPP TEMPLATE command.	3-81
3-18	Parameters displayed in the output of the SHOW PPP TEMPLATE DEBUG command.	3-83
3-19	Parameters displayed in the output of the SHOW PPP TXSTATUS command.	3-84
4-1	S/T loop transmission states defined by ITU-T Recommendation I.430.	4-5
4-2	Categories of debug messages generated by the BRI software module.	4-8
4-3	Standard LAPD configuration for an ISDN Basic Rate Interface.	4-13
4-4	Standard LAPD configuration for an ISDN Primary Rate Interface.	4-13
4-5	SAPI values used by LAPD to specify types of layer 3 entities.	4-14
4-6	TEI values used by LAPD to specify logical devices attached to a Basic Rate Interface.	4-14
4-7	TEI values used by LAPD to specify logical devices attached to a Primary Rate Interface.	4-14
4-8	LAPD frame types.	4-15
4-9	SPID Initialisation States.	4-19
4-10	SPID Initialisation Events.	4-20
4-11	SPID File States.	4-20
4-12	SPID File Events.	4-21
4-13	Automatic Switch Detection States.	4-21
4-14	Automatic Switch Detection Events.	4-22
4-15	Call priority and call bumping.	4-25
4-16	Example configuration parameters for AODI.	4-30
4-17	Example configuration parameters for a basic ISDN network.	4-33
4-18	Q.931 Profiles.	4-35
4-19	ISDN Basic Rate Interface conformance tests.	4-60
4-20	ISDN Basic Rate Interface debug options.	4-62
4-21	ISDN Basic Rate Interface test modes for S/T interfaces using an MC145474 controller.	4-62
4-22	ISDN Basic Rate Interface test modes for S/T interfaces using a PSB2186 controller.	4-63
4-23	ISDN Basic Rate Interface test modes for U interfaces using a PEB2091 controller.	4-63
4-24	ISDN Basic Rate Interface test modes for U interfaces using an MC145572 controller.	4-64
4-25	Parameters displayed in the output of the ENABLE Q931 DEBUG=MDECODE command.	4-68
4-26	Parameters displayed in the output of the ENABLE Q931 DEBUG=MRAW command.	4-68
4-27	Parameters displayed in the output of the ENABLE Q931 DEBUG=SDLC command.	4-69
4-28	Parameters displayed in the output of the ENABLE Q931 DEBUG=SSPID command.	4-69
4-29	Parameters displayed in the output of the ENABLE Q931 DEBUG=SSPIDFILE command.	4-70
4-30	Parameters displayed in the output of the ENABLE Q931 DEBUG=STATE command.	4-71
4-31	Q.931 Profiles.	4-85
4-32	Parameters displayed in the output of the SHOW BRI CONFIGURATION command.	4-87
4-33	Parameters displayed in the output of the SHOW BRI COUNTERS=INTERFACE command.	4-88
4-34	Parameters displayed in the output of the SHOW BRI COUNTERS=BRI command.	4-92
4-35	Parameters displayed in the output of the SHOW BRI CTEST command.	4-94
4-36	Parameters displayed in the output of the SHOW BRI DEBUG command.	4-96

4-37	Parameters displayed in the output of the SHOW BRI STATE command for an S/T interface.	4-97
4-38	States of the physical layer state machine for an ISDN Basic Rate S/T Interface.	4-97
4-39	Parameters displayed in the output of the SHOW BRI STATE command for a U interface.	4-98
4-40	States of the physical layer state machine for an ISDN Basic Rate U Interface.	4-99
4-41	ISDN Basic Rate Interface test modes for S/T interfaces using an MC145474 controller.	4-100
4-42	ISDN Basic Rate Interface test modes for S/T interfaces using a PSB2186 controller.	4-101
4-43	ISDN Basic Rate Interface test modes for U interfaces using a PEB2091 controller.	4-102
4-44	ISDN Basic Rate Interface test modes for U interfaces using an MC145572 controller.	4-103
4-45	Parameters displayed in the output of the SHOW ISDN CALL command.	4-104
4-46	Parameters displayed in the output of the SHOW ISDN CALL command for a specified call name.	4-105
4-47	Parameters displayed in the output of the SHOW ISDN CLIST command.	4-108
4-48	Parameters displayed in the output of the SHOW ISDN LOG command.	4-109
4-49	Parameters displayed in the output of the SHOW LAPD command.	4-111
4-50	Parameters displayed in the output of the SHOW LAPD COUNT command.	4-113
4-51	Parameters displayed in the output of the SHOW LAPD STATE command.	4-114
4-52	Parameters displayed in the output of the SHOW Q931 command.	4-116
4-53	Parameters displayed in the output of the SHOW Q931 CALL command.	4-117
4-54	Parameters displayed in the output of the SHOW Q931 SPID command.	4-118
4-55	Parameters displayed in the output of the SHOW Q931 SPID command during the auto-SPID procedure.	4-120
5-1	NLPID values for protocol encapsulation over X.25 circuits.	5-5
5-2	Parameters displayed in the output of the SHOW MIOX command.	5-24
5-3	Parameters displayed in the output of the SHOW MIOX COUNT command.	5-25
5-4	Parameters displayed in the output of the SHOW MIOX CIRCUIT command.	5-27
5-5	Parameters displayed in the output of the SHOW MIOX CIRCUIT COUNTER command.	5-28
5-6	Parameters displayed in the output of the SHOW MIOX CIRCUIT ENCAP command.	5-29
5-7	Parameters displayed in the output of the SHOW X25T command.	5-31
5-8	Parameters displayed in the output of the SHOW X25T CIRCUIT command.	5-32
5-9	Parameters displayed in the output of the SHOW X25T COUNT command.	5-33
5-10	Parameters displayed in the output of the SHOW X25T CPAR command.	5-35
6-1	Functions of the fields in an IP datagram.	6-5
6-2	Internet Protocol address classes and limits on numbers of networks and hosts.	6-6
6-3	ICMP messages implemented by the router.	6-12
6-4	TOS values defined by RFC 1349.	6-16
6-5	Example configuration parameters for broadcast forwarding to a unicast address.	6-27
6-6	Example configuration parameters for broadcast forwarding to a multicast address.	6-29
6-7	Example configuration parameters for a basic TCP/IP network.	6-34
6-8	Example configuration parameters for IP filtering.	6-39
6-9	Predefined port names used by the IP filtering process.	6-46
6-10	Predefined ICMP type names used by the IP filtering process.	6-47
6-11	Predefined ICMP code names used by the IP filtering process.	6-48
6-12	Predefined protocol names used by the IP filtering process.	6-49
6-13	DHCP reply parameters used by the router for configuring IP.	6-54
6-14	Parameters displayed in the output of the SHOW BOOTP RELAY command.	6-106
6-15	Parameters displayed in the output of the SHOW IP command.	6-107
6-16	Parameters displayed in the output of the SHOW IP ARP command.	6-109
6-17	Parameters displayed in the output of the SHOW IP COUNTER=ICMP command.	6-110
6-18	Parameters displayed in the output of the SHOW IP COUNTER=INTERFACE command.	6-111
6-19	Parameters displayed in the output of the SHOW IP COUNTER=IP command.	6-112
6-20	Parameters displayed in the output of the SHOW IP COUNTER=MULTICAST command.	6-114
6-21	Parameters displayed in the output of the SHOW IP COUNTER=RIP command.	6-114
6-22	Parameters displayed in the output of the SHOW IP COUNTER=ROUTE command.	6-115
6-23	Parameters displayed in the output of the SHOW IP COUNTER=UDP command.	6-116
6-24	Parameters displayed in the output of the SHOW IP FILTER command.	6-118
6-25	Parameters displayed in the output of the SHOW IP HELPER command.	6-119
6-26	Parameters displayed in the output of the SHOW IP HELPER COUNTER command.	6-120

6-27	Parameters displayed in the output of the SHOW IP HOST command.	6-121
6-28	Parameters displayed in the output of the SHOW IP INTERFACE command.	6-122
6-29	Parameters displayed in the output of the SHOW IP INTERFACE COUNTER command.	6-123
6-30	Parameters displayed in the output of the SHOW IP POOL command.	6-125
6-31	Parameters displayed in the output of the SHOW IP RIP command.	6-126
6-32	Parameters displayed in the output of the SHOW IP RIPTIMER command.	6-127
6-33	Parameters displayed in the output of the SHOW IP RIP COUNTER command.	6-129
6-34	Parameters displayed in the output of the SHOW IP ROUTE command.	6-131
6-35	Parameters displayed in the output of the SHOW IP ROUTE GENERAL command.	6-132
6-36	Parameters displayed in the output of the SHOW IP ROUTE CACHE command.	6-132
6-37	Parameters displayed in the output of the SHOW IP ROUTE COUNT command.	6-133
6-38	Parameters displayed in the output of the SHOW IP ROUTE FILTER command.	6-134
6-39	Parameters displayed in the output of the SHOW IP ROUTE TEMPLATE command.	6-135
6-40	Parameters displayed in the output of the SHOW IP ROUTE TEMPLATE command.	6-135
6-41	Parameters displayed in the output of the SHOW IP UDP command.	6-137
6-42	Parameters displayed in the output of the SHOW PING command.	6-138
6-43	Parameters displayed in the output of the SHOW TCP command for a specified TCP connection.	6-140
6-44	Parameters displayed in the output of the SHOW TCP command.	6-142
6-45	TCP states.	6-142
6-46	Parameters displayed in the output of the SHOW TRACE command.	6-144
7-1	Configuration parameters for TTY devices.	7-3
7-2	Command line editing functions and keystrokes.	7-4
7-3	Parameters displayed in the output of the SHOW TTY command.	7-9
7-4	Parameters displayed in the output of the SHOW TTY=ALL SUMMARY command.	7-10
7-5	Parameters displayed in the output of the SHOW TTY DEFAULT command.	7-10
8-1	Suggested software compression speed settings for different transmission line speeds.	8-8
8-2	Parameters displayed in the output of the SHOW ENCO command.	8-9
8-3	Parameters displayed in the output of the SHOW ENCO CHANNEL command.	8-9
8-4	Parameters displayed in the output of the SHOW ENCO CHANNEL command for a specified channel.	8-10
8-5	Parameters displayed in the output of the SHOW CHANNELS COUNTERS command.	8-12
8-6	Parameters displayed in the output of the SHOW ENCO COUNTERS=JOBPROCESSING command.	8-15
8-7	Parameters displayed in the output of the SHOW ENCO COUNTERS=STAC command.	8-16
8-8	Parameters displayed in the output of the SHOW ENCO COUNTERS=USER command.	8-19
8-9	Parameters displayed in the output of the SHOW ENCO COUNTERS=UTIL command.	8-21
9-1	Possible test outcomes for an Ethernet interface.	9-5
9-2	Possible test outcomes for an asynchronous interface.	9-7
9-3	Possible test outcomes for a Basic Rate ISDN interface.	9-7
9-4	Valid interface options for the DISABLE TEST INTERFACE command.	9-8
9-5	Valid interface options for the ENABLE TEST INTERFACE command.	9-9
9-6	Parameters displayed in the output of the SHOW TEST INTERFACE command.	9-11
9-7	Parameters displayed in the output of the SHOW TEST INTERFACE COUNTERS command.	9-12
10-1	Parameters displayed in the output of the SHOW TRIGGER command.	10-17
10-2	Parameters displayed in the output of the SHOW TRIGGER FULL command.	10-18
10-3	Parameters displayed in the output of the SHOW TRIGGER STATUS command.	10-19
10-4	Parameters displayed in the output of the SHOW TRIGGER COUNTER command.	10-20
11-1	Parameters displayed in the output of the SHOW TDM GROUP command.	11-7
12-1	Log message fields.	12-3
12-2	Log message severity levels.	12-4
12-3	Mapping between logging facility module identifier, type and subtype, and syslog facility identifiers.	12-7

12-4	Mapping between logging facility severity levels and syslog levels.	12-7
12-5	Log message filter comparison operators.	12-9
12-6	Example configuration parameters for a basic logging facility.	12-10
12-7	Recognised time zone names.	12-18
12-8	Parameters displayed in the output of the SHOW LOG command.	12-35
12-9	Parameters displayed in the output of the SHOW LOG FULL command.	12-37
12-10	Parameters displayed in the output of the SHOW LOG COUNTERS command.	12-38
12-11	Parameters displayed in the output of the SHOW LOG OUTPUT command.	12-41
12-12	Parameters displayed in the output of the SHOW LOG OUTPUT FULL command.	12-42
12-13	Parameters displayed in the output of the SHOW LOG QUEUE command.	12-44
12-14	Parameters displayed in the output of the SHOW LOG RECEIVE command.	12-46
12-15	Parameters displayed in the output of the SHOW LOG STATUS command.	12-47
13-1	Parameters displayed in the output of the SHOW SCRIPT command.	13-12
14-1	Call priority and call bumping.	14-6
14-2	Affect of BCAP and HLC parameters on ISDN call SETUP messages.	14-11
14-3	PBX debugging options.	14-19
14-4	Parameters displayed in the output of the SHOW PBX command.	14-26
14-5	Parameters displayed in the output of the SHOW PBX CALL command.	14-26
14-6	Parameters displayed in the output of the SHOW PBX EXTENSION command.	14-28
14-7	Parameters displayed in the output of the SHOW PBX GROUP command.	14-29
15-1	Parameters displayed in the output of the SHOW DHCP command.	15-22
15-2	Parameters displayed in the output of the SHOW DHCP CLIENT command.	15-24
15-3	Parameters displayed in the output of the SHOW DHCP POLICY command.	15-25
15-4	Parameters displayed in the output of the SHOW DHCP RANGE command.	15-26
16-1	Access modes for MIB objects.	16-5
16-2	Status values for MIB objects.	16-6
16-3	Fields in an SNMP message.	16-7
16-4	SNMP PDUs.	16-7
16-5	Generic SNMP traps.	16-7
16-6	Community profiles for objects in a MIB view.	16-8
16-7	Parameters displayed in the output of the SHOW SNMP command.	16-20
16-8	Parameters displayed in the output of the SHOW SNMP COMMUNITY command.	16-23
17-1	Log types and subtypes for firewall events.	17-6
17-2	Predefined IP protocol service names.	17-15
17-3	Parameters displayed in the output of the SHOW FIREWALL command.	17-25
17-4	Parameters displayed in the output of the SHOW FIREWALL POLICY command.	17-27
17-5	Parameters displayed in the output of the SHOW FIREWALL POLICY COUNTERS command.	17-30
17-6	Parameters displayed in the output of the SHOW FIREWALL SESSION command.	17-33
B-1	Module Identifiers, Names and Descriptions.	B-2
B-2	FLASH File System Message Codes.	B-4
B-3	ISDN Q.931 Call Clearance Cause Codes and Descriptions.	B-6
B-4	Log Message Types and Subtypes.	B-8
C-1	MIBs supported by the AR Router.	C-2
C-2	Object identifiers for Allied Telesyn AR router products.	C-4
C-3	Object groups in the AT Router sub-tree of the Allied Telesyn Enterprise MIB.	C-4
C-4	Object identifiers for AR Router base CPU and expansion boards.	C-5
C-5	Object identifiers for AR Router interface types.	C-5
C-6	Object identifiers for AR Router chip sets.	C-6
C-7	MIB-II implementation variations.	C-12
C-8	Host Resources MIB implementation variations.	C-15
C-9	Host Resources MIB device types supported by the router.	C-16

Preface

Purpose of this Manual

This manual is the complete reference to the configuration, management and operation of the AR100 Series Internet Router, and includes detailed descriptions of all management commands.

The AR100 Series Internet Router, provides efficient and cost-effective multiprotocol routing, terminal serving and integrated network management over wide area networks and LANs. All models can run the same software and can provide all of the following functions simultaneously (depending on the hardware configuration):

- Wide area networking via Point-to-Point Protocol and X.25.
- Basic Rate access to Integrated Services Digital Network (ISDN) services, with dial-on-demand and channel aggregation.
- TCP/IP and RIP routing protocols.
- ARP and Proxy ARP address resolution protocols.
- Sophisticated packet filtering.
- Van Jacobson's header compression and STAC LZS compression.
- Terminal serving using Telnet, with local host nicknames.
- Sophisticated, configurable event logging facility for network monitoring and alarm notification to single or multiple management centres.
- Triggers for automatic and timed execution of commands in response to events.
- Scripting for automated configuration of routers and centralised management of configurations.
- PBX services for models fitted with both analogue voice interfaces.
- Dynamic Host Configuration Protocol (DHCP) for automatically assigning IP addresses and other configuration information to PCs and other hosts on TCP/IP networks.
- Support for the Simple Network Management Protocol (SNMP), standard MIBs and the Allied Telesyn Enterprise MIB, enabling the router to be managed by a separate SNMP management station.
- A stateful inspection firewall.

Intended Audience

This manual is intended for the system administrator, network manager or communications technician who will configure and maintain the AR100 Series Internet Router, or who manages a network of AR routers.

It is assumed that the reader is familiar with:

- The topology of the network in which the AR100 router is to be used.
- Basic principles of computer networking, protocols and routing, and interfaces.
- Administration and operation of a computer network.

This manual is not intended for users who will use the computer network to access network services from their terminal, personal computer or workstation. Most of the commands described in this manual require MANAGER privilege and can only be entered from a terminal or port which has been assigned MANAGER privilege.

Structure of this Manual

This manual is organised into the following chapters:

- *Command Summary* is an alphabetical list of all router commands and their syntax.
- *Chapter 1, Operation* describes general operation, management and support features, including user authentication, down-line loading and installing software releases.
- *Chapter 2, Interfaces* describes the Ethernet synchronous and asynchronous network interfaces on the router.
- *Chapter 3, Point-to-Point Protocol (PPP)* describes the router's implementation of the Point-to-Point Protocol (PPP).
- *Chapter 4, Integrated Services Digital Network (ISDN)* describes the ISDN service provided by the router, and how to configure ISDN interfaces.
- *Chapter 5, X.25* describes how to configure the router's implementation of the ITU-T Recommendation X.25 protocol, and how to build an X.25 Packet Switched Network.
- *Chapter 6, Internet Protocol (IP)* describes the router's implementation of the Internet Protocol (IP).
- *Chapter 7, Terminal Server* describes the terminal services provided by the router, and the router's implementation of the Internet Telnet protocol.
- *Chapter 8, Compression Services* describes the data compression and encryption services provided by the router.
- *Chapter 9, Test Facility* describes the facilities built into the router for testing the router's interfaces, and how to execute and interpret the tests.
- *Chapter 10, Trigger Facility* describes the router's trigger facility for automated and timed execution of management commands in response to events.
- *Chapter 11, Time Division Multiplexing (TDM)* describes the router's implementation of time division multiplexing over G.703 links.

- *Chapter 12, Logging Facility* describes the router's advanced logging facility and how to configure the logging facility to provide flexible monitoring of the router's activities.
- *Chapter 13, Scripting* describes the router's scripting facility for creating, storing and executing sequences of commands.
- *Chapter 14, Telephony Services* describes the router's comprehensive PBX services for models fitted with both analogue voice and ISDN interfaces.
- *Chapter 15, Dynamic Host Configuration Protocol (DHCP)* describes the router's implementation of the Dynamic Host Configuration Protocol.
- *Chapter 16, Simple Network Management Protocol (SNMP)* describes the router's implementation of the Simple Network Management Protocol.
- *Chapter 17, Firewall* describes the router's packet filtering firewall using stateful inspection.
- *Chapter 18, Link Compression* describes the link compression and encryption facilities provided by the router for Point-to-Point Protocol (PPP), Frame Relay and X.25 links.
- *Appendix A, Messages* provides a complete listing of all the informational, warning and error messages generated by the router.
- *Appendix B, Reference Tables* provides reference tables of identifiers and return codes for a range of router functions and network services.
- *Appendix C, SNMP MIBs* describes the *Management Information Bases (MIBs)* and managed objects supported by the router's SNMP agent, including the Allied Telesyn Enterprise MIB.
- *Glossary* contains definitions of terms and concepts used throughout this manual.
- *Index* is a master index to topics and commands covered in this manual.

Related Manuals

This manual is part of the AR100 Series Internet Router Documentation Set. The complete AR100 Series Internet Router Documentation Set comprises:

- *AR100 Series Internet Router Start Here Guide.*
- *AR100 Series Internet Router User Guide.*
- *AR100 Series Internet Router Reference Manual.*

Other Documents

Supported Standards and Protocols

Table I on page xxxii lists the protocols and standards supported by the AR100 router and the references where these protocols and standards are defined.

Table I: Protocols and standards supported by the AR router.

Protocol/standard	Reference
ARP	RFCs 826, 925.
Assigned Numbers	RFC 1700.
DHCP	RFCs 1541, 1542.
ICMP	RFCs 792, 950.
IEEE 802.2	ANSI/IEEE Std 802.2-1985.
IEEE 802.3	ANSI/IEEE Std 802.3-1985, 802.3a, b, c, e-1988.
HTTP	RFCs 1521, 1945.
IP	RFCs 791, 821, 950, 951, 1009, 1055, 1122, 1144, 1349, 1542, 1812, 1858.
IP addressing	RFC 1597.
IP Security Associations	RFCs 1825, 1827, 1829.
ISDN	ANSI T1.231-1997, ANSI T1.403-1995, ANSI T1.408-1990, AT&T TR 54016-1989, Austel TS 013.1:1990, Bellcore SR-3887 1997, TS 013.2:1990, TS 014.1:1990, TS 014.2:1990; ITU G.703, ITU G.704, ITU G.706, ITU-T Recommendations G.703 (1972), ITU-T Recommendation Q.922, G.794 (1988), G.706 (1988), I.120 (1988), I.121 (1988), I.411 (1988), I.430 (1988), I.431 (1988), Q.920 (1988), Q.921 (1988), Q.930 (1988), Q.931 (1988); ETSI Specifications ETS 300 011:1991, ETS 300 012:1992, ETS 300 102-1:1990, ETS 300 1022:1990, ETS 300 125:1991, ETS 300 153:1992, ETS 300 156:1992; New Zealand Telecom TNA 134; German Monopol (BAPT 221); Japan NTT I.430-a, Rockwell Bt8370 Fully Intergrated T1/E1 Framer and Line Interface data sheet, Technical Reference of Frame Relay Interface, Ver. 1, November 1993, Nippon Telegraph and Telephone Corporation.
MIOX	RFC 1356.
NAT	RFC 1631.
Point-to-Point Protocol	RFCs 1331–1334, 1376, 1378, 1548, 1549, 1552, 1570, 1638, 1661, 1662, 1762, 1962, 1968, 1974, 1978, 1990, 2125.
Proxy ARP	RFC 1027.
RIP	RFCs 1058, 1388.
SNMP, MIBs	RFCs 1155, 1157, 1213, 1239, 1315, 1398, 1493, 1514, 1573, 2233.
TCP	RFC 793.
Telnet	RFCs 854–858, 932 1091.
TFTP	RFC 1350.
UDP	RFC 768.
Van Jacobson's compression	RFC 1144.
X.25	ITU-T Recommendations X.25 (1988), X.121 (1988).

Obtaining Copies of Internet Protocols and Standards

The Internet Protocols are defined in *Requests For Comments* (RFCs). RFCs are developed and published under the auspices of the *Internet Engineering Steering Group* (IESG) of the *Internet Engineering Task Force* (IETF). For more information about the IESG and IETF, visit the IETF web site at <http://www.ietf.org/>. For more information about RFCs and Internet Drafts (the starting point for RFCs), visit the RFC Editor web site at <http://www.rfc-editor.org/>. This site has information about the RFC standards process, archives of RFCs and current Internet Drafts, links to RFC indexes and search engines, and a list of other RFC repositories.

RFCs can be obtained electronically from many RFC repositories, mail servers, World Wide Web (WWW), Gopher or WAIS sites. A good starting point for finding the nearest RFC repository is to point your Web browser at <http://www.isi.edu/in-notes/rfc-retrieval.txt>.

To obtain a copy of an RFC using FTP, FTP to the host and login as user `anonymous`, and a password of either `guest` or your email address. The FTP server will usually prompt you for one or the other. Use the `get` command to retrieve the desired RFC. Most sites have a file, usually `rfc-index.txt`, which lists the titles and file names of all available RFCs. Most sites have a file, usually `rfc-retrieval.txt`, which gives detailed information about RFC repositories and how to retrieve RFCs via FTP, mail servers, WWW, Gopher and WAIS.

To learn how to obtain a copy of an RFC via email from a mail server, point your browser at <http://www.isi.edu/in-notes/rfc-editor/rfc-info>.

To obtain a copy of an RFC from a Web site, or to search RFC repositories for a specific RFC or all RFCs relating to a topic, point your Web browser at <http://www.rfc-editor.org/rfc.html>.

Background Reading

For an introduction to the Internet Protocols refer to:

DDN Protocol Handbook, Elizabeth J. Feinler, 1991, DDN Network Information Center, SRI International, 333 Ravenswood Avenue, Menlo Park, CA 94025, USA. Email: nic@nic.ddn.mil.

Internetworking with TCP/IP — Volume I: Principles, protocols and architecture (2nd Edition), Douglas E. Comer, 1991, Prentice-Hall International, Inc., New Jersey. ISBN 0-13-474321-0.

Internetworking with TCP/IP — Volume II: Design, implementation, and internals, Douglas E. Comer and David L. Stevens, 1991, Prentice-Hall International, Inc., New Jersey. ISBN 0-13-472242-6.

Internetworking with TCP/IP — Volume III: Client-server programming and applications, Douglas E. Comer and David L. Stevens, 1993, Prentice-Hall International, Inc., New Jersey. ISBN 0-13-474222-2.

For a description of layered protocols refer to:

Computer networks (2nd Edition), Andrew S. Tanenbaum, 1989, Prentice-Hall International, Inc., New Jersey. ISBN 0-13-162959-0.

For an introduction to network management refer to:

The simple book — An introduction to management of TCP/IP-based Internets,
Marshall T. Rose, 1991, Prentice-Hall International, Inc. ISBN 013812611-9.

For an introduction to firewalls and internet security refer to:

Firewalls and Internet Security — Repelling the Wily Hacker, William R.
Cheswick and Steven M. Bellovin, Addison-Wesley Publishing Company,
Reading, Massachusetts. ISBN 0-201-63357-4.

These books are listed for your convenience. Allied Telesyn does not endorse these books or recommend them over others on the same subject.

Publicly Accessible Documents

Allied Telesyn maintains an online archive of documents and files that customers can access via the World Wide Web or via anonymous FTP. For WWW access, point your Web browser at <http://www.alliedtelesyn.com> or <http://www.alliedtelesyn.co.nz>. For access via anonymous FTP, FTP to host [ftp.alliedtelesyn.co.nz](ftp://ftp.alliedtelesyn.co.nz), login as user `anonymous` and enter your email address (e.g. `username@host.org`) when prompted for a password.

Conventions

A number of symbols, typographic and stylist conventions are used throughout this manual to aid learning and make information easier to find (Table II on page xxxv).

Table II: Typographic conventions used in this manual.

This typeface	Is used for
<i>Italic</i>	Referring to another section in this manual or another manual, or to introduce and emphasise new terms. For example, "See <i>Chapter 2, Configuration</i> ".
Monospace	Text as it appears on-screen, or anything you must type.
0xFF	Numbers starting with the 0x prefix are hexadecimal values.
[Key]	A key on your keyboard. For example, "at the prompt, type a command and press [Enter]. Example key names include [Shift], [Alt], [Ctrl] and [Backspace].
[Key/Key]	A pair of keys on your keyboard that should be pressed together. For example, [Ctrl/P] means "press and hold down the [Ctrl] key, and while holding down the [Ctrl] key, press and release the [P] key, then release the [Ctrl] key".
[Key,Key]	<p>A sequence of keys that should be pressed in sequence. For example, [Break,T] means "press and release the [Break] key, then press and release the [T] key".</p> <p>The [Key/Key] and [Key,Key] symbols may be combined, as in [Ctrl/P,T], which means "press and hold down the [Ctrl] key, and while holding down the [Ctrl] key, press and release the [P] key, release the [Ctrl] key, then press and release the [T] key".</p>
Attention	A special keystroke known as the attention character, which will be either [Break] or [Ctrl/P].



Note. A note like this presents additional information or interesting sidelights.



Warning. A warning alerts you to situations in which you could do something that might result in a loss of data, or cause damage to the equipment.

Screen views show examples of the output resulting from particular commands, or what the screen should look like at a particular time, for instance:

Configuration for ETH instance 0:

Module	Protocol	Format	Discrim	MAC address
IPG	IP	Ethernet	0800	0000cd000027
IPG	ARP	Ethernet	0806	0000cd000027
IPX	Novell	Novell	–	0000cd000027
DNT	DECnet	Ethernet	6003	aa0004003908
Bridging	LAT	Ethernet	6004	–
Bridging	EtherTalk	SNAP	00000080f3	–

Commands are described under *Command Reference* within the section to which they apply. Command syntax is defined using these conventions:

This	Is used for														
CAPS	Keywords to be typed as shown. In general keywords may be abbreviated to the shortest string that is unambiguous within the current context. The exception is commands with a profound effect, such as RESTART IMMEDIATELY, which must be typed in full.														
<i>italic</i>	A variable placeholder, to be replaced by an actual value in a command.														
[]	Square brackets enclose optional items. Enter the item or items required, but do not type the brackets.														
	Vertical bars separate choices in a list — choose one of the items.														
...	Ellipses indicate that the preceding element may be repeated any number of times.														
{ }	Braces surround a required choice of options; you must choose one of the options listed.														
n..m	Defines a range of values from n to m inclusive. n and m are decimal numbers.														
<i>interface</i>	An interface type — one of: <table> <tr> <td>ETHn</td><td>for Ethernet interfaces</td></tr> <tr> <td>PORTn</td><td>for Asynchronous interfaces</td></tr> <tr> <td>BRI n</td><td>for Basic Rate ISDN interfaces</td></tr> <tr> <td>PPPn</td><td>for Point-to-Point interfaces</td></tr> <tr> <td>LAPDn</td><td>for LAPD interfaces</td></tr> <tr> <td>X25Tn</td><td>for X.25 DTE interfaces</td></tr> <tr> <td>n</td><td>when defining one of the above interface types. n is a non-negative, zero-based decimal number.</td></tr> </table>	ETHn	for Ethernet interfaces	PORTn	for Asynchronous interfaces	BRI n	for Basic Rate ISDN interfaces	PPPn	for Point-to-Point interfaces	LAPDn	for LAPD interfaces	X25Tn	for X.25 DTE interfaces	n	when defining one of the above interface types. n is a non-negative, zero-based decimal number.
ETHn	for Ethernet interfaces														
PORTn	for Asynchronous interfaces														
BRI n	for Basic Rate ISDN interfaces														
PPPn	for Point-to-Point interfaces														
LAPDn	for LAPD interfaces														
X25Tn	for X.25 DTE interfaces														
n	when defining one of the above interface types. n is a non-negative, zero-based decimal number.														
<i>ipadd</i>	An IP address in dotted decimal form (e.g. 131.203.9.197). In some situations an address in domain name format.														
<i>macadd</i>	A hardware address (such as an Ethernet address) of the form xxxxxxxxxxxx, where xx is a two-digit hexadecimal number with leading zeros if necessary.														

Allied Telesyn Offices and Locations

UNITED KINGDOM

Tel: (+44) 1235 442500
Freephone: 0800 204040
Fax: (+44) 1235 442590

EIRE

Tel: 1 800 409 127

SWEDEN

Tel: 08 131414

NORWAY

Tel: 2211 1181

DENMARK

Tel: 3332 3006

FINLAND

Tel: 0800 98040

FRANCE

Tel: (+33) 01 60 92 15 25
Fax: (+33) 01 69 28 37 49

BELGIUM

Tel: (+32) 2 481 60 60
Fax: (+32) 2 463 17 06

GERMANY

Toll Free: 00 800 255 43310
Tel: (+49) 30 435 90 00
Fax: (+49) 30 435 706 50

GERMANY - SOUTH

Tel: (+49) 8161 99 060
Fax: (+49) 8161 99 0622

EAST EUROPE - AUSTRIA

Tel: (+43) 1 8762441
Fax: (+43) 1 8762572

ITALY

Tel: (+39) 02 416047
Fax: (+39) 02 419282

SPAIN

Tel: (+34) 91 5591055
Fax: (+34) 91 5592644

U.S.A.

Toll Free: 1-800-424-4284
Tech Support: 1-800-428-4835
Fax: (425) 489-9191
<http://www.alliedtelesyn.com>

CANADA

Tel: (905) 709-7444
Fax: (905) 709-7400

Latin America

Tel: 1-425-481-3852
Fax: 1-425-489-9191

Singapore

Tel: (+65) 383-3832
Fax: (+65) 383-3830

Australia

Tel: (+61) 2-9438-5111
Fax: (+61) 2-9438-496

NEW ZEALAND

Tel: (+64) 3 377 8900
Fax: (+64) 3 377 8870
<http://www.alliedtelesyn.co.nz>

Safety and Statutory Information

Safety and Statutory Statements

This information must be read prior to use of this equipment, and overrides as appropriate any information in respect of connection and use of the equipment.

Any enquiries regarding regulatory aspects of this equipment should be addressed to Allied Telesyn International.

This product meets the requirements of EN60950 1992 including amendments 1,2 and 3, AS3260 and Austel TS001.

1. Connection to Mains Voltage Supply

Products supplied for connection to mains voltage must only be used with the supplied mains lead.

The wires in the mains lead are coloured in accordance with the following code:

GREEN and YELLOW	EARTH
BLUE	NEUTRAL
BROWN	LIVE

For continued protection against the risk of fire and shock hazard, replace fuses only with the same type and rating.

2. Earthing

The power cord supplied with this equipment must be connected to a power socket which provides a reliable protective earth connection.

3. Ports for the Connection of Other Apparatus

The following interfaces normally operate at SELV (Safe Extra Low Voltage) levels:

- Asynchronous (console) ports.
- Hub ports.



SELV is a voltage that does not exceed 42.4V peak a.c. or 60V d.c.

The following interfaces are Telecommunications Network Voltage (TNV) circuits which operate normally within the limits of SELV:

- BRI ports for connection to ISDN Basic Rate telecommunications networks.

The following interfaces are Telecommunications Network Voltage (TNV2) circuits:

- POTS ports (AR140 only).

4. Product Servicing

This product contains no user-serviceable parts. All product servicing must be carried out by qualified service personnel.

5. Lithium Cell

This product includes an Integrated Circuit containing a Lithium cell. These devices can be identified by one of the part numbers DS1286, DS1213C, DS1213D or CD2032SLF, and the warning statement:

Do not dispose in fire

This warning should be strictly adhered to. Do not attempt to open this device. Disposal of this device must be carried out by qualified service personnel.

CAUTION

Danger of explosion if the IC is incorrectly replaced. Replace only with the same type or equivalent type recommended by the manufacturer. Dispose of used IC according to the manufacturer's instructions.

6. National Variants

Norway

The local distributor of the AR Router in Norway must attach a self adhesive label placed just above the fuse rating, which is situated above the mains inlet filter. This label displays the following text in Norwegian.

Apparatet må kun tilkoples jordet stikkontakt.

Sweden

The local distributor of the AR Router in Sweden must attach a self adhesive label placed just above the fuse rating, which is situated above the mains inlet filter. This label displays the following text in Swedish.

Apparaten skall anslutas till jordat uttag när den ansluts till ett nätverk.

Switzerland

The local distributor of the AR Router in Switzerland must supply a moulded plug that conforms to SEV / ASE 1011.

Denmark

The local distributor of the AR Router in Denmark must ensure that the power supply cord is provided with a moulded plug.

7. Sicherheitsinformation

Wir raten Ihnen, bevor Sie dieses Gerät in Deutschland benutzen, diese Information zu lesen.

Alle andere informationen sind von dieser information über die Anschlüsse und die Anwendung des Geräts außer Kraft gesetzt worden.

7.1 Anschluß an das allgemeine Stromnetz

Die Geräte, die an das allgemeine Stromnetz anschlossen werden, sollen nur mit dem gelieferten Kabeln benutzt werden. Die Sicherheitssteckdose ist in der Nähe des Geräts installiert werden und leicht zu erreichen sein.

Die Drähte in der Netzleitung sind in den folgenden Farben gekennzeichnet

GRÜN und GELB	ERDUNG
BLAU	NEUTRAL
BRAUN	LEITER

Für weiterreichenden Schutz gegen Feuerrisiko und die Möglichkeiten eines elektrischen Schocks zu verlangen, sollen Sicherungen nur durch solche derselbe Typs und derselben Stärke ersetzt werden.

8. Erdung

Die Erdung soll durch die versorgten netzkabel geliefert worden.

9. Sicherheit des Leitungsnetz

Alle Anschlüsse, die im Format RJ45,15 and 37 Anschlusstypen D-sind, haben die Sicherheitstellung 'SELV CIRCUIT'.

10. Wartung des Geräts

Diese Gerät erhält keinen Bestandteile, die vom Benutzer selbst gewartet werden können

Versuche von nicht qualifiziertem Personal, Zugang zum Gerateinneren zu erlangen, kompromittieren die Zulassungsbedingungen. Im Falle eines solchen versuchten Zugangs wird keine Haftung dafür übernommen, wenn sich das Gerät als nicht mehr den Zulassungsbedingungen übereinstimmend erweist.

Alle Produktwartung muß von qualifizierten Servicepersonnell durch geführt werden.

11. Lithiumzelle

Diese Productt einschließt einen integrierte Schaltkreis, der einen Lithiumzelle enthält. Man kann dieses Organ mit dem Teilnummer DS1286 und DS1213D, und die folgende Warnung identifizieren:

werden sie nicht diese zelle in feuer los

Man sollen bei dieser Warnung bleiben. Man soll nicht versuchen, diese Organ zu öffnen. Es ist nur die qualifizierten Servicepersonnell, die diese Zelle loswerden sollen.

12. EMC Compliance

This product meets the requirements of the European Electromagnetic Compatibility (EMC) Directive 89/336/EEC. The product complies with the requirements of CISPR 22 (EN55022) for Emissions and EN50082-1 for Immunity for limits of radio disturbance characteristics for Information Technology Equipment (ITE).

13. Intended Use

This equipment is not intended for use in a computer room as defined in the standard for the Protection of Electronic Computer/Data Processing Equipment ANSI/NFPA 75 relating to North America only.

14. US Federal Communications Commission (FCC)

This equipment complies with Part 15 of FCC Rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

Properly shielded and grounded cables and connectors must be used in order to meet FCC emission limits. Proper cables and connectors are available from Allied Telesyn authorised distributors and resellers. Allied Telesyn is not responsible for any radio or television interference caused by using other than recommended cables and connectors or by unauthorised changes or modifications to this equipment. Unauthorised changes or modifications could void the user's authority to operate the equipment.

16. Canadian Department of Communications

This digital apparatus does not exceed the limits for radio noise emissions from digital apparatus as set out in the Radio Interface Regulations of the Canadian Department of Communications.

Le present appareil numerique n'émet pas de bruits radioelectriques dépassant les limites applicables aux appareils numeriques prescrites dans le Reglement sur le brouillage radioelectrique edicte par le ministere des Communications du Canada.

17. Telepermit PTC 231/97/027

The VOX ports on AR300 Series routers comply with specification PTC 100 for connection to the New Zealand Telecom Network.

Not all standard phones/faxes etc. will respond to incoming ringing. This may not be a fault, and could be due to incompatibility between the TA and the CPE. Check with a phone which is known to ring before reporting a fault.

Telecom "Smart" services which are accessed via switch-hook flash, will not be accessible to CPE connected to the TA.

Environmental Conditions

This apparatus is designed for operation under the following environmental conditions:

Operating Temperature	-5°C to 40°C
Storage Temperature	-5°C to 70°C
Humidity	5% to 95% non condensing
Operating Atmospheric Pressure	86 kPa to 106 kPa

Reader's Comments

Please tell us how we can improve our manuals. Please complete this form and return it to us.

How do you rate this manual?	Excellent	Good	Average	Fair	Poor
Accuracy of information	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Completeness of information	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clarity of writing	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Organisation of information	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ease of finding information	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Number of figures	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Usefulness of figures	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Number of examples	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Usefulness of examples	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Usefulness of the Index	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

What did you like most about this manual?

What did you like least about this manual?

What would you like to see more of?

What would you like to see less of?

Did you find any errors in this manual?

Page	Description
<hr/>	<hr/>
<hr/>	<hr/>
<hr/>	<hr/>
<hr/>	<hr/>
<hr/>	<hr/>

Any additional comments or suggestions to improve this manual? _____

What computer software or hardware product has the best manual you have used?

Product _____ Manufacturer _____

Why? _____

fold here

What is your experience with routers?

☐ No previous experience

☐ Less than 1 year's experience

☐ More than 1 year's experience on one model

☐ More than 1 year's experience on more than one model

What Software version are you using? (from the display of the SHOW SYSTEM command)

Software Version: _____ Release Version: _____

Patch Installed: _____

Name/Title	_____	Date	_____
Company/Dept	_____		
Address	_____	Phone	_____
	_____	Fax	_____
	_____		_____
	_____		_____

fold here

Please post this form to:

**Allied Telesyn International
PO Box 10-290
Christchurch
New Zealand**

Attention: Reader's Comments

Command Summary

ACTIVATE FLASH COMPACTION	1-26
ACTIVATE ISDN CALL=name	4-42
ACTIVATE MIOX=x25t-interface CIRCUIT=circuit-name [USER=IP]	5-9
ACTIVATE PPP=ppp-interface RXPKT=hexstring	3-27
ACTIVATE Q931=interface ASPID	4-43
ACTIVATE Q931=interface MESSAGE=message [DLC=dlc-index]	4-43
ACTIVATE SCRIPT=filename [OUTPUT=device] [<i>parameters</i>]	13-5
ACTIVATE TRIGGER=trigger-id	10-5
ADD ALIAS=name STRING=substitution	1-28
ADD BOOTP RELAY=ipadd	6-43
ADD DHCP POLICY=name [ARPTIMEOUT=seconds] [BOOTFILESIZE=bootfilesize] [BROADCASTADDRESS=ipadd] [COOKIESERVER=ipadd,ipadd...] [DNSSERVER=ipadd,ipadd...] [DOMAINNAME=string] [ETHERENCAP={ON OFF}] [EXTENSIONPATH=string] [FILE=string] [HOSTNAME=string] [IMPRESSSERVER=ipadd,ipadd...] [INTMTU=mtu] [IPFORWARDING={ENABLED DISABLED}] [IPMTU=mtu] [IPPLATEAU=mtu,mtu...] [IPTIMEOUT=seconds] [IPTTL=ttl] [LOGSERVER=ipadd,ipadd...] [LPRSERVER=ipadd,ipadd...] [MASKDISCOVERY={ON OFF}] [MASKSUPPLIER={ON OFF}] [MERITDUMPFIL=string] [NAMESERVER=ipadd,ipadd...] [NBDDSERVERS=ipadd,ipadd...] [NBNAMESEVER=ipadd,ipadd...] [NBNODETYPE={BNODE PNODE MNODE HNODE}] [NBSCOPE=string] [NISDOMAIN=string] [NISERVERS=ipadd,ipadd...] [NTPSERVERS=ipadd,ipadd...] [POLICYFILTERING=ipadd,ipadd...] [RESOURCESERVER=ipadd,ipadd...] [ROOTPATH=string] [ROUTER=ipadd,ipadd...] [ROUTERDISCOVERY={ON OFF}] [ROUTERSOLICIT=ipadd] [SERVER=ipadd] [SERVERNAME=server-name] [SOURCEROUTING={ENABLED DISABLED}] [STATICROUTE=ipadd,ipadd...] [SUBLOCAL={ON OFF}] [SUBNETMASK=ipadd] [SWAPSERVER=ipadd] [T1TIME=seconds] [T2TIME=seconds] [TCPGARBAGE={ON OFF}] [TCPKEEPALIVE=seconds] [TCPTTL=ttl] [TIMEOFFSET=utc-offset] [TIMESERVER=ipadd,ipadd...] [TRAILERENCAP={ON OFF}] [XDISPLAYSERVERS=ipadd,ipadd...] [XFONTSERVERS=ipadd,ipadd...]	15-4
ADD DHCP RANGE=name IP=ipadd ADDRESS=macadd [POLICY=name]	15-9
ADD FIREWALL POLICY=name INTERFACE=interface TYPE={PUBLIC PRIVATE} [METHOD={DYNAMIC PASSALL}]	17-11
ADD FIREWALL POLICY=name NAT={ENHANCED STANDARD} INTERFACE=interface [IP=ipadd] GBLINTERFACE=interface [GBLIP=ipadd[-ipadd]]	17-12
ADD FIREWALL POLICY=name RULE=rule-id ACTION={ALLOW DENY} INTERFACE=interface PROTOCOL={protocol ALL EGP GRE OSPF SA TCP UDP} [GBLIP=ipadd] [GBLPORT={ALL port[-port]}] [IP=ipadd[-ipadd]] [PORT={ALL port[-port] service-name} [REMOTEIP=ipadd[-ipadd]] [SOURCEPORT={ALL port[-port]}]	17-14
ADD IP ARP=ipadd INTERFACE=interface {CIRCUIT=miox-circuit ETHERNET=macadd}	6-44

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

ADD IP FILTER=filter-number SOURCE=ipadd [SMASK=ipadd] [SPORT={port-name|port-id}]
 [DESTINATION=ipadd [DMASK=ipadd]] [DPORT={port-name|port-id}] [ICMPCODE={icmp-code-name|
 icmp-code-id}] [ICMPTYPE={icmp-type-name|icmp-type-id}] [LOG={4..1600|DUMP|HEADER|NONE}]
 [OPTIONS={YES|NO}] [PROTOCOL={protocol|ANY|EGP|ICMP|OSPF|TCP|UDP}] [SESSION={ANY|ESTABLISHED|
 START}] [SIZE=size] [ENTRY=entry-number]
 {ACTION={INCLUDE|EXCLUDE}} [POLICY=0..15] [PRIORITY=P0..P7] 6-45

ADD IP HELPER DESTINATION=ipadd INTERFACE=interface PORT=port-number 6-51

ADD IP HOST=name IPADDRESS=ipadd 6-52

ADD IP INTERFACE=interface IPADDRESS={ipadd|DHCP} [BROADCAST={0|1}] [DIRECTEDBROADCAST={YES|NO|
 ON|OFF}] [FILTER={0..99|NONE}] [FRAGMENT={YES|NO}] [MASK=ipadd] [METRIC=1..16] [MULTICAST={OFF|
 SEND|RECEIVE|BOTH|ON}] [POLICYFILTER={100..199|NONE}] [PRIORITYFILTER={200..299|NONE}]
 [PROXYARP={ON|OFF}] [RIPMETRIC=1..16] [VJC={ON|OFF}] 6-53

ADD IP RIP INTERFACE=interface [CIRCUIT=miox-circuit] [IP=ipadd] [SEND={NONE|RIP1|RIP2|COMPATIBLE}]
 [RECEIVE={NONE|RIP1|RIP2|BOTH}] [DEMAND={NO|YES}] [AUTH={NONE|PASSWORD|MD5}]
 [PASSWORD=password] 6-56

ADD IP ROUTE FILTER={filter-id} IP=ipadd MASK=ipadd ACTION={INCLUDE|EXCLUDE} [DIRECTION={RECEIVE|
 SEND|BOTH}] [INTERFACE=interface] [NEXTHOP=ipadd] [POLICY=0..7]
 [PROTOCOL={ANY|RIP|STATIC|INTERFACE}] 6-59

ADD IP ROUTE TEMPLATE=name INTERFACE=interface NEXTHOP=ipadd [CIRCUIT=miox-circuit]
 [METRIC=1..16] [METRIC1=1..16] [POLICY=0..7] [PREFERENCE=0..65535] 6-60

ADD IP ROUTE=ipadd INTERFACE=interface NEXTHOP=ipadd [CIRCUIT=miox-circuit] [MASK=ipadd]
 [METRIC=1..16] [METRIC1=1..16] [POLICY=0..7] [PREFERENCE=0..65535] 6-57

ADD IP TRUSTED=ipadd 6-61

ADD ISDN CALL=name NUMBER=number PRECEDENCE={IN|OUT} [ALTNUMBER=number]
 [BUMPDELAY=0..100] [CALLBACK={ON|OFF|YES|NO|TRUE|FALSE}] [CALLINGNUMBER=number]
 [CALLINGSUBADDRESS=calling-subaddress] [CBDELAY=0..100] [CHECKCLI={OFF|PRESENT|REQUIRED}]
 [CHECKSUB={OFF|LOCAL|REMOTE}] [CHECKUSER={OFF|LOCAL|REMOTE}] [CLLIST=0..99] [DIRECTION={IN|
 OUT|BOTH}] [DOV={ON|OFF|YES|NO|TRUE|FALSE}] [HOLDUP=0..7200] [INANY={ON|OFF|YES|NO|TRUE|
 FALSE}] [INTPREF={NONE|interface}] [INTREQ={NONE|interface}] [KEEPU={ON|OFF|YES|NO|TRUE|FALSE}]
 [LOGIN={ALL|NONE|CHAP|PAP-RADIUS|PAP-TACACS|USER}] [OUTCLI={OFF|CALLING|INTERFACE|
 NONUMBER}] [OUTSUB={OFF|LOCAL|REMOTE}] [OUTUSER={OFF|LOCAL|REMOTE}] [PASSWORD={NONE|
 CLI|CALLED|SUB|NAME|USER}] [PPPTemplate=template] [PRIORITY=0..99] [RATE={56K|64K}]
 [REMOTECALL=name|remote-number] [RN1=0..10] [RN2=0..5] [RT1=5..120] [RT2=300..1200]
 [SEARCHCLI={ON|OFF|YES|NO|TRUE|FALSE|CALLED|0..99}] [SEARCHSUB={OFF|LOCAL|REMOTE}]
 [SEARCHUSER={OFF|LOCAL|REMOTE}] [SUBADDRESS=number] [USER={ATTACH|PPP}] [USERNAME={NONE|
 CLI|CALLED|SUB|NAME|USER}] 4-44

ADD ISDN CLLIST=0..99 NUMBER=number 4-50

ADD ISDN DOMAINNAME=domain-name 4-50

ADD LAPD=interface TEI=tei... 4-51

ADD LAPD=interface XSPID=spid-index 4-51

ADD LAPD=interface XTEI=tei 4-52

ADD LOG OUTPUT={TEMPORARY|output-id} [FILTER=filter-id] [ACTION={PROCESS|IGNORE}] [ALL]
 [DATE={op}dd-mmm-yyyy] [DEVICE={op}device] [FILE={op}filename] [MASK=ipadd] [MSGTEXT={op}string]
 [MODULE={op}module-id] [ORIGIN=ipadd] [REFERENCE={op}string] [SEVERITY={op}severity]
 [SOURCELINE={op}line] [SUBTYPE={op}subtype-id] [TIME={op}hh:mm:ss] [TYPE={op}type-id] 12-12

ADD LOG RECEIVE={ipadd|ANY} [ALLOW={YES|NO}] [MASK=ipadd] [PASSWORD={password|NONE}]
 [PROTOCOL={ALL|BOTH|NEW|OLD|SYSLOG}] 12-15

ADD MIOX=x25t-interface CIRCUIT=circuit-name {DTEADDRESS=dteaddress|PVC=1..4095} [CPAR=0..8]
 [ENCAP={IP|NULL|MULTIPLE}] [COMMENT=comment] [COMP={ON|OFF}] [TCPCOMP={ON|OFF}] 5-10

ADD PPP=ppp-interface OVER=physical-interface [AUTHENTICATION={CHAP|EITHER|PAP|NONE}]
 [AUTHMODE={IN|OUT|INOUT}] [CBDELAY=1..100] [CBMODE={ACCEPT|OFF|REQUEST}]
 [CBNUMBER=e164number] [CBOperation={E164NUMBER|USERAUTH}] [COMPALGORITHM=STACLSZS]
 [COMPRESSION={LINK|OFF}] [CONFIGURE={value|CONTINUOUS}] [LQR={ON|OFF|time}] [MAGIC={ON|OFF}]
 [NUMBER=number] [RESTART=time] [STACHECK={LCB|SEQUENCE}] [TERMINATE={value|CONTINUOUS}]
 [TYPE={DEMAND|PRIMARY|SECONDARY}] 3-28

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

ADD SCRIPT=filename TEXT=text [LINE=line]	13-6
ADD SNMP COMMUNITY=name [TRAPHOST=ipadd] [MANAGER=ipadd]	16-13
ADD TDM GROUP=groupname SLOTS=slotlist	11-3
ADD TRIGGER=trigger-id SCRIPT=filename... [NUMBER=index]	10-6
ADD USER=login-name PASSWORD=password [CBNUMBER=e164number] [DESCRIPTION=description] [PRIVILEGE={USER MANAGER}] [TELNET={YES NO}]	1-28
ADD X25T CPAR=call-index [COPY=call-index] [MAXDATA={128 256 512 1024}] [NUI=nui] [RMAXDATA={128 256 512 1024}] [RWINDOW=1..127] [TMAXDATA={128 256 512 1024}] [TWINDOW=1..127] [USERDATA=hex-string] [WINDOW=1..127]	5-11
CLEAR FLASH TOTALLY	1-29
CREATE CONFIG=filename	1-30
CREATE DHCP POLICY=name LEASETIME={lease-time INFINITY} [INHERIT=name]	15-9
CREATE DHCP RANGE=name POLICY=name IP=ipadd NUMBER=number [GATEWAY=ipadd]	15-10
CREATE FFILE=filename {DATA=bytes ADDRESS=address LENGTH=length}	1-31
CREATE FIREWALL POLICY=name	17-16
CREATE IP POOL=pool-name IP=ipadd[-ipadd]	6-62
CREATE LOG OUTPUT={TEMPORARY output-id} DESTINATION={MEMORY PORT ROUTER SYSLOG} [FORMAT={FULL MSGONLY SUMMARY}] [MAXQUEUESEVERITY=severity] [MESSAGES=message-count] [PASSWORD={password NONE}] [PORT=port-number] [QUEUEONLY={YES NO}] [SECURE={YES NO}] [SERVER=ipadd] [ZONE={time-zone-name utc-offset}]	12-16
CREATE PBX EXTENSION=extension-number [BCAP={SPEECH AUDIO}][CALLINGNUMBER={calling-number OFF}][COPY=extension-number] [HLC={DEFAULT FAX TELEPHONE}] [GROUP=group-name] [NAME=extension-name] [NOHLC={ACCEPT REJECT}] [NUMACCEPT={matching-number ALL NOTPRESENT OFF}] [PORT={pbx-interface NONE}] [SUBACCEPT={matching-subaddr ALL NOTPRESENT OFF}] [SUPPRESS={1..30 NONE}] [TERMINATE={0..30 NONE}]	14-13
CREATE PBX GROUP=group-name [EXTENSION=extension-number] [HUNT={SEARCH NONE}] [NUMACCEPT={matching-number ALL NOTPRESENT OFF}] [SUBACCEPT={matching-subaddr ALL NOTPRESENT OFF}]	14-16
CREATE PPP=ppp-interface OVER=physical-interface [AUTHENTICATION={CHAP EITHER PAP NONE}] [AUTHMODE={IN OUT INOUT}] [BAP={ON OFF}] [BAPMODE={CALL CALLBACK}] [CBDELAY=1..100] [CBMODE={ACCEPT OFF REQUEST}] [CBNUMBER=e164number] [CBOperation={E164NUMBER USERAUTH}] [COMPALGORITHM=STACLSZS] [COMPRESSION={ON OFF LINK}] [CONFIGURE={value CONTINUOUS}] [DEBUGMAXBYTES=16..256] [DESCRIPTION=description] [DOWNRATE=0..100] [DOWNTIME=time] [ECHO={ON OFF period}] [FRAGMENT={ON OFF}] [FRAGOVERHEAD=0.100] [IDLE={ON OFF time}] [INDATALIMIT={NONE 1..65535}] [IPPOOL={pool-name NONE}] [IPREQUEST={ON OFF}] [LQR={ON OFF time}] [MAGIC={ON OFF}] [NULLFRAGTIMER=time] [NUMBER=number] [ONLINELIMIT={NONE 1..65535}] [OUTDALIMIT={NONE 1..65535}] [PASSWORD=password] [RESTART=time] [STACHECK={LCB SEQUENCE}] [TERMINATE={value CONTINUOUS}] [TOTALDALIMIT={NONE 1..65535}] [TYPE={DEMAND PRIMARY SECONDARY}] [UPRATE=0..100] [UPTIME=time] [USERNAME=username]	3-31
CREATE SNMP COMMUNITY=name [ACCESS={READ WRITE}] [TRAPHOST=ipadd] [MANAGER=ipadd] [OPEN={ON OFF YES NO TRUE FALSE}]	16-14
CREATE TDM GROUP=groupname INTERFACE=interface SLOTS=slotlist	11-4
CREATE TRIGGER=trigger-id [CPU=value] [DIRECTION={UP DOWN ANY}] [INTERFACE=interface EVENT={UP DOWN FAIL ANY}] [CIRCUIT=miox-circuit] [CP={BCP CCP PCP LCP}] [MEMORY=value] [DIRECTION={UP DOWN ANY}] [PERIODIC=minutes] [REBOOT={RESTART CRASH ALL}] [TIME=hh:mm] [DATE=date] [DAYS=day-list] [AFTER=hh:mm] [BEFORE=hh:mm] [SCRIPT=filename...] [NAME=name] [REPEAT={YES NO ONCE FOREVER count}] [STATE={ENABLED DISABLED}] [TEST={YES NO ON OFF}]	10-7
CREATE X25T=x25-interface OVER=LAPDn [MAXACTIVE=0..4095] [MODULUS={8 128}] [T20=1..360][T21=1..360] [T22=1..360] [T23=1..360] [T24={1..360 OFF}] [T27={1..360 OFF}] [MINRECALL=1..360] [R20=0..65535] [R22=0..65535] [R23=0..65535] [R27=0..65535] [NPVC=0..4095] [DEFCPAR=0..8] [DTEADDRESS=dteaddress] [LIC=0..4095] [HIC=0..4095] [LTC=0..4095] [HTC=0..4095] [LOC=0..4095] [HOC=0..4095] [ROLE={DYNAMIC DCE DTE}]	5-12
DEACTIVATE ISDN CALL={acnum name}	4-52

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

DEACTIVATE MIOX=x25t-interface CIRCUIT=circuit-name [USER=IP]	5-14
DEACTIVATE SCRIPT=filename	13-7
DELETE ALIAS=name	1-31
DELETE BOOTP RELAY=ipadd	6-63
DELETE DHCP POLICY=name [ARPTIMEOUT] [BOOTFILESIZE] [BROADCASTADDRESS] [COOKIESERVER] [DNSSERVER] [DOMAINNAME] [ETHERENCAP] [EXTENSIONPATH] [FILE] [HOSTNAME] [IMPRESSSERVER] [INTMTU] [IPFORWARDING] [IPMTU] [IPPLATEAU] [IPTIMEOUT] [IPTTL] [LOGSERVER] [LPRSERVER] [MASKDISCOVERY] [MASKSUPPLIER] [MERITDUMPFIL] [NAMESERVER] [NBDDSERVERS] [NBNAMESEVER] [NBNODETYPE] [NBScope] [NISDOMAIN] [NISERVERS] [NTPSERVERS] [POLICYFILTERING] [RESOURCESERVER] [ROOTPATH] [ROUTER] [ROUTERDISCOVERY] [ROUTERSOLICIT] [SERVER] [SERVERNAME] [SOURCEROUTING] [STATICROUTE] [SUBLOCAL] [SUBNETMASK] [SWAPSERVER] [T1TIME] [T2TIME] [TCPGARBAGE] [TCPKEEPALIVE] [TCPTTL] [TIMEOFFSET] [TIMESERVER] [TRAILERENCAP] [XDISPLAYSERVERS] [XFONTSERVERS]	15-11
DELETE DHCP RANGE=name IP=ipadd	15-15
DELETE FFIL=filename	1-32
DELETE FILE=filename	1-32
DELETE FIREWALL POLICY=name INTERFACE=interface	17-17
DELETE FIREWALL POLICY=name NAT={ENHANCED STANDARD} INTERFACE=interface GBLINTERFACE=interface [IP=ipadd]	17-18
DELETE FIREWALL POLICY=name RULE=rule-id	17-19
DELETE FIREWALL SESSION={session-number ALL}	17-19
DELETE INSTALL={TEMPORARY PREFERRED DEFAULT}	1-33
DELETE IP ARP=ipadd	6-63
DELETE IP FILTER=filter-number ENTRY={entry-number ALL}	6-64
DELETE IP HELPER DESTINATION=ipadd INTERFACE=interface PORT=port-number	6-64
DELETE IP HOST=name	6-65
DELETE IP INTERFACE=interface	6-66
DELETE IP RIP INTERFACE=interface [CIRCUIT=miox-circuit] [IP=ipadd]	6-67
DELETE IP ROUTE FILTER=filter-id	6-69
DELETE IP ROUTE TEMPLATE=name	6-69
DELETE IP ROUTE=ipadd MASK=ipadd INTERFACE=interface NEXTHOP=ipadd	6-68
DELETE IP TRUSTED=ipadd	6-69
DELETE ISDN CALL=name	4-53
DELETE ISDN CLILIST=0..99 NUMBER=number	4-53
DELETE ISDN DOMAINNAME[=domain-name]	4-54
DELETE LAPD=interface TEI=tei	4-54
DELETE LAPD=interface XSPID=spid-index	4-55
DELETE LAPD=interface XTEI=tei	4-55
DELETE LOG OUTPUT={TEMPORARY output-id} FILTER={ALL filter-id}	12-19
DELETE LOG RECEIVE={ipadd ANY}	12-20
DELETE MIOX=x25t-interface CIRCUIT=circuit-name	5-15
DELETE PPP=ppp-interface OVER=physical-interface [NUMBER=number] [TYPE={DEMAND PRIMARY SECONDARY}]	3-41
DELETE SCRIPT=filename [LINE=line]	13-8
DELETE SNMP COMMUNITY=name [TRAPHOST=ipadd] [MANAGER=ipadd]	16-15
DELETE TCP=tcb	6-70
DELETE TDM GROUP=groupname SLOTS=slotlist	11-5
DELETE TRIGGER=trigger-id NUMBER=index	10-11
DELETE USER=login-name	1-33
DELETE X25T CPAR=call-index	5-15
DESTROY DHCP POLICY=name	15-15

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

DESTROY DHCP RANGE=name	15-16
DESTROY FIREWALL POLICY=name	17-20
DESTROY IP POOL=pool-name	6-71
DESTROY LOG OUTPUT={TEMPORARY output-id}	12-20
DESTROY PBX EXTENSION=extension-number	14-17
DESTROY PBX GROUP=group-name	14-18
DESTROY PPP TEMPLATE=template	3-42
DESTROY PPP=ppp-interface	3-41
DESTROY SNMP COMMUNITY=name	16-16
DESTROY TDM GROUP=groupname	11-5
DESTROY TRIGGER=trigger-id	10-11
DESTROY X25T=x25-interface	5-16
DISABLE BOOTP RELAY	6-71
DISABLE BRI=instance CTEST	4-56
DISABLE BRI=instance TEST[=test-number]	4-57
DISABLE BRI[=instance] DEBUG[={ERRORS INDICATIONS STATES EVENTS ALL}]	4-56
DISABLE DHCP [BOOTP]	15-16
DISABLE ENCO COMPSTATISTICS	8-5
DISABLE ENCO DEBUGGING=PACKET	8-6
DISABLE FIREWALL	17-20
DISABLE FIREWALL POLICY=name [DEBUG={ALL PACKET PKT PROCESS}] [ICMP_FORWARDING={ALL PARAMETER PING SOURCEQUENCH TIMEEXCEEDED TIMESTAMP UNREACHABLE}] [LOG={ALLOW DENY DENYDUMP INAICMP INALLOW INAOTHER INATCP INAUDP INDDICMP INDDOTHER INDDTCP INDDUDP INDDUMP INDENY INDICMP INDOTHER INDTCP INDUDP OUTAICMP OUTALLOW OUTAOTHER OUTATCP OUTAUDP OUTDDICMP OUTDDOTHER OUTDDTCP OUTDDUDP OUTDDUMP OUTDENY OUTDICMP OUTDOTHER OUTDTCP OUTDUDP}] [OPTIONS={ALL RECORD_ROUTE SECURITY SOURCEROUTE TIMESTAMP}] [PING]	17-21
DISABLE HTTP DEBUG={ALL AUTH MSG SESSION}	1-34
DISABLE HTTP DEBUG={ALL AUTH MSG SESSION}	1-40
DISABLE HTTP SERVER	1-34
DISABLE INTERFACE={ifIndex interface DYNAMIC} LINKTRAP	2-13
DISABLE IP	6-72
DISABLE IP DEBUG	6-72
DISABLE IP DNSRELAY	6-73
DISABLE IP ECHOREPLY	6-73
DISABLE IP FOFILTER	6-73
DISABLE IP FORWARDING	6-74
DISABLE IP HELPER	6-74
DISABLE IP INTERFACE=interface	6-75
DISABLE IP REMOTEASSIGN	6-75
DISABLE IP SRCROUTE	6-76
DISABLE ISDN CALL=name	4-58
DISABLE ISDN LOG	4-58
DISABLE LOG	12-21
DISABLE LOG GENERATION	12-21
DISABLE LOG OUTPUT[={TEMPORARY output-id}]	12-21
DISABLE LOG RECEPTION	12-22
DISABLE MIOX=x25t-interface CIRCUIT=circuit-name	5-16
DISABLE PBX DEBUG={ALL CODEC COMMAND COUNTERS CLID EVENT REDIRECTEDNUMBER TRACE}	14-18
DISABLE PORT=port-number	2-14

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

DISABLE PPP TEMPLATE=template DEBUG={ALL AUTH BAPPKT BAPSTATE CALLBACK DEMAND ENCO LCP NCP PKT UTILISATION}[,...]	3-43
DISABLE PPP=ppp-interface	3-42
DISABLE PPP=ppp-interface DEBUG={ALL AUTH BAPPKT BAPSTATE CALLBACK DEMAND ENCO LCP NCP PKT UTILISATION}[,...]	3-43
DISABLE Q931=interface DEBUG={MDECODE MRW SDLC SINTERFACE SSPID SSPIDFILE STATE TRACE}	4-59
DISABLE RCAP	4-60
DISABLE SNMP	16-16
DISABLE SNMP AUTHENTICATE_TRAP	16-16
DISABLE SNMP COMMUNITY=name [TRAP]	16-17
DISABLE TEST INTERFACE=interface	9-8
DISABLE TRIGGER[=trigger-id]	10-12
DISABLE USER=login-name	1-35
DUMP [ADDR=address] [LEN=length] [SIZE={BYTE LONG WORD}] [SPACE={SD SP UD UP UR}]	1-35
EDIT [filename]	1-37
ENABLE BOOTP RELAY	6-77
ENABLE BRI=instance CTEST=test-number	4-60
ENABLE BRI=instance TEST=test-number	4-62
ENABLE BRI=[instance] DEBUG={ERRORS INDICATIONS STATES EVENTS ALL}	4-61
ENABLE DHCP [BOOTP]	15-17
ENABLE ENCO COMPSTATISTICS	8-6
ENABLE ENCO DEBUGGING=PACKET	8-7
ENABLE FIREWALL	17-22
ENABLE FIREWALL POLICY=name [DEBUG={ALL PACKET PKT PROCESS}] [ICMP_FORWARDING={ALL PARAMETER PING SOURCEQUENCH TIMEEXCEEDED TIMESTAMP UNREACHABLE}] [LOG={ALLOW DENY DENYDUMP INAICMP INALLOW INAOOTHER INATCP INAUDP INDDICMP INDDOTHER INDDTC INDDUDP INDDUMP INDENY INDICMP INDOTHER INDTCP INDUDP OUTAICMP OUTALLOW OUTAOOTHER OUTATCP OUTAUDP OUTDDICMP OUTDDOTHER OUTDDTCP OUTDDUDP OUTDDUMP OUTDENY OUTDICMP OUTDOTHER OUTDTC OUTDUDP}] [OPTIONS={ALL RECORD_ROUTE SECURITY SOURCEROUTE TIMESTAMP}] [PING]	17-22
ENABLE HTTP SERVER	1-40
ENABLE INTERFACE={ifIndex interface DYNAMIC} LINKTRAP	2-14
ENABLE IP	6-77
ENABLE IP DEBUG	6-78
ENABLE IP DNSRELAY	6-78
ENABLE IP ECHOREPLY	6-78
ENABLE IP FOFILTER	6-78
ENABLE IP FORWARDING	6-79
ENABLE IP HELPER	6-80
ENABLE IP INTERFACE=interface	6-80
ENABLE IP REMOTEASSIGN	6-81
ENABLE IP ROUTE {CACHE COUNT MULTIPATH}	6-76
ENABLE IP ROUTE {CACHE COUNT MULTIPATH}	6-81
ENABLE IP SRCROUTE	6-82
ENABLE ISDN CALL=name	4-65
ENABLE ISDN LOG	4-65
ENABLE LOG	12-22
ENABLE LOG GENERATION	12-23
ENABLE LOG OUTPUT[={TEMPORARY output-id}]	12-23
ENABLE LOG RECEPTION	12-23
ENABLE MIOX=x25t-interface CIRCUIT=circuit-name	5-17

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

ENABLE PBX DEBUG={ALL CODEC COMMAND COUNTERS CLID EVENT REDIRECTEDNUMBER TRACE} [PORT= <i>port-number</i>]	14-19
ENABLE PORT= <i>port-number</i>	2-15
ENABLE PPP TEMPLATE=template DEBUG={ALL AUTH BAPPKT BAPSTATE CALLBACK DEMAND ENCO LCP NCP PKT UTILISATION}[,...] [PORT= <i>port-number</i>] [TIMEOUT={NONE 1..4000000000}] [NUMPKTS={CONT 1..4000000000}]	3-46
ENABLE PPP= <i>ppp-interface</i>	3-44
ENABLE PPP= <i>ppp-interface</i> DEBUG={ALL AUTH BAPPKT BAPSTATE CALLBACK DEMAND ENCO LCP NCP PKT UTILISATION}[,...] [PORT= <i>port-number</i>] [TIMEOUT={NONE 1..4000000000}] [NUMPKTS={CONT 1..4000000000}]	3-45
ENABLE Q931= <i>interface</i> ASPID= <i>index[,index...]</i>	4-66
ENABLE Q931= <i>interface</i> DEBUG={MDECODE MRW SDLC SINTERFACE SSPID SSPIDFILE STATE TRACE} .	4-66
ENABLE RAPI	4-71
ENABLE SNMP	16-17
ENABLE SNMP AUTHENTICATE_TRAP	16-18
ENABLE SNMP COMMUNITY= <i>name</i> [TRAP]	16-18
ENABLE TEST INTERFACE= <i>interface</i> [TIME= <i>time</i> CONT] [MORE]	9-9
ENABLE TRIGGER[= <i>trigger-id</i>]	10-12
ENABLE USER= <i>login-name</i>	1-41
FLUSH LOG OUTPUT[={TEMPORARY <i>output-id</i> }]	12-24
HELP [<i>topic</i>]	1-41
IF <i>string1</i> {EQ NE} <i>string2</i> THEN <i>commands</i> [ELSE <i>commands</i>] ENDIF	13-8
LOAD [METHOD={HTTP WEB WWW}] [DELAY= <i>delay</i>] [DESTINATION=FLASH] [FILE= <i>filename</i>] [PROXYPORT=1..65535] [SERVER= <i>ipadd</i>]	1-42
LOAD [METHOD=NONE] [DELAY= <i>delay</i>] [DESTINATION=FLASH] [FILE= <i>filename</i>] [PORT= <i>port</i>]	1-42
LOAD [METHOD=TFTP] [DELAY= <i>delay</i>] [DESTINATION=FLASH] [FILE= <i>filename</i>] [SERVER= <i>ipadd</i>]	1-42
LOAD [METHOD=ZMODEM] [DELAY= <i>delay</i>] [DESTINATION=FLASH] [PORT= <i>port</i>]	1-42
LOGIN [<i>login-name</i>]	1-44
LOGOFF	1-45
MODIFY ADDR= <i>address</i> SIZE={BYTE LONG WORD} VAL= <i>value-list</i> [SPACE={SD SP UD UP UR}]	1-45
PING [[IPADDRESS= <i>ipadd</i>] [DELAY= <i>seconds</i>] [LENGTH= <i>number</i>] [NUMBER={ <i>number</i> CONTINUOUS}] [PATTERN= <i>hexnum</i>] [SIPADDRESS= <i>ipadd</i>] [SCREENOUTPUT={YES NO}] [TIMEOUT= <i>number</i>] [TOS= <i>number</i>]	6-82
PURGE BOOTP RELAY	6-84
PURGE IP	6-84
PURGE LOG[={TEMPORARY <i>output-id</i> }]	12-24
PURGE PORT={ <i>port-number</i> ALL}	2-15
PURGE PPP	3-47
PURGE TDM GROUP	11-6
PURGE TRIGGER	10-13
PURGE USER	1-46
RENAME <i>src-filename</i> <i>dest-filename</i>	1-46
RESET BRI= <i>instance</i>	4-72
RESET BRI[= <i>instance</i>] COUNTERS[={INTERFACE BRI}]	4-72
RESET ENCO COUNTERS={JOBPROCESSING STAC USER UTIL}	8-7
RESET ETH= <i>n</i>	2-16
RESET ETH[= <i>n</i>] COUNTERS[={COLLISION DIAGNOSTIC DOT3STAT INTERFACE}]	2-16
RESET HTTP SERVER	1-47
RESET IP	6-84
RESET IP COUNTER={ALL GENERAL ICMP INTERFACE ROUTE UDP}	6-85
RESET IP INTERFACE= <i>interface</i>	6-85

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

RESET LOADER	1-47
RESET PORT[=port-number]	2-17
RESET PORT[=port-number] COUNTERS[={DIAGNOSTIC INTERFACE RS232}]	2-17
RESET PORT[=port-number] HISTORY	2-18
RESET PPP=ppp-interface [COUNTERS] [LINKCOUNTERS={ONLINE INDATA OUTDATA TOTALDATA ALL}]	3-48
RESET Q931=interface [CALL={call-index ALL}]	4-73
RESET TEST INTERFACE	9-10
RESET USER[=login-name] [COUNTERS[={ALL GLOBAL USER}]]	1-48
RESET X25T=x25-interface	5-18
RESTART {REBOOT ROUTER} [CONFIG=filename NONE]	1-49
SET [TIME=time] [DATE=date]	1-56
SET BOOTP MAXHOPS=1..16	6-86
SET BRI=instance [ACTIVATION={NORMAL ALWAYS}] [ISDNSLOTS=slot-list] [MODE={ISDN TDM MIXED}] [TDMLOTS=slot-list]	4-73
SET CONFIG=filename	1-49
SET DHCP POLICY=name [ARPTIMEOUT=seconds] [BOOTFILESIZE=bootfilesize] [BROADCASTADDRESS=ipadd] [COOKIESERVER=ipadd,ipadd...] [DNSSERVER=ipadd,ipadd...] [DOMAINNAME=string] [ETHERENCAP={ON OFF}] [EXTENSIONPATH=string] [FILE=string] [HOSTNAME=string] [IMPRESSSERVER=ipadd,ipadd...] [INTMTU=mtu] [IPFORWARDING={ENABLED DISABLED}] [IPMTU=mtu] [IPPLATEAU=mtu,mtu...] [IPTIMEOUT=seconds] [IPTTL=ttl] [LOGSERVER=ipadd,ipadd...] [LPRSERVER=ipadd,ipadd...] [MASKDISCOVERY={ON OFF}] [MASKSUPPLIER={ON OFF}] [MERITDUMPFIL=string] [NAMESEVER=ipadd,ipadd...] [NBDDSERVERS=ipadd,ipadd...] [NBNAMESEVER=ipadd,ipadd...] [NBNODETYPE={BNODE PNODE MNODE HNODE}] [NBSCOPE=string] [NISDOMAIN=string] [NISERVERS=ipadd,ipadd...] [NTPSERVERS=ipadd,ipadd...] [POLICYFILTERING=ipadd,ipadd...] [RESOURCESEVER=ipadd,ipadd...] [ROOTPATH=string] [ROUTER=ipadd,ipadd...] [ROUTERDISCOVERY={ON OFF}] [ROUTERSOLICIT=ipadd] [SERVER=ipadd] [SERVERNAME=server-name] [SOURCEROUTING={ENABLED DISABLED}] [STATICROUTE=ipadd,ipadd...] [SUBLOCAL={ON OFF}] [SUBNETMASK=ipadd] [SWAPSERVER=ipadd] [T1TIME=seconds] [T2TIME=seconds] [TCPGARBAGE={ON OFF}] [TCPKEEPALIVE=seconds] [TCPTTL=ttl] [TIMEOFFSET=utc-offset] [TIMESERVER=ipadd,ipadd...] [TRAILERENCAP={ON OFF}] [XDISPLAYSERVERS=ipadd,ipadd...] [XFONTSERVERS=ipadd,ipadd...]	15-17
SET ENCO SW [STACCHANNELS=0..4] [STACSPED=0..3]	8-7
SET FIREWALL POLICY=name RULE=rule-id [PROTOCOL={protocol ALL EGP GRE OSPF SA TCP UDP}] [GBLIP=ipadd] [GBLPORT={ALL port[-port]}] [IP=ipadd[-ipadd]] [PORT={ALL port[-port] service-name}] [REMOTEIP=ipadd[-ipadd]] [SOURCEPORT={ALL port[-port]}]	17-24
SET HELP=helpfile	1-50
SET INSTALL={TEMPORARY PREFERRED DEFAULT} [RELEASE={release-name EPROM}]	1-50
SET INTERFACE={ifIndex interface DYNAMIC} TRAPLIMIT=1..60	2-18
SET IP ARP=ipadd INTERFACE=interface [CIRCUIT=miox-circuit ETHERNET=macadd]	6-86
SET IP AUTONOMOUS=1..65535	6-87
SET IP FILTER=filter-number ENTRY=entry-number [SOURCE=ipadd] [SMASK=ipadd] [SPORT={port-name port-id}] [DESTINATION=ipadd] [DMASK=ipadd] [DPORT={port-name port-id}] [ICMPCODE={icmp-code-name icmp-code-id}] [ICMPSTYPE={icmp-type-name icmp-type-id}] [LOG={4..1600 DUMP HEADER NONE}] [OPTIONS={YES NO}] [PROTOCOL={protocol ANY EGP ICMP OSPF TCP UDP}] [SESSION={ANY ESTABLISHED START}] [SIZE=size] [ACTION={INCLUDE EXCLUDE}] [POLICY=0..15] [PRIORITY=P0..P7]	6-88
SET IP HOST=name IPADDRESS=ipadd	6-91
SET IP INTERFACE=interface [BROADCAST={0 1}] [DIRECTEDBROADCAST={YES NO ON OFF}] [FILTER={0..99 NONE}] [FRAGMENT={YES NO}] [IPADDRESS=ipadd DHCP] [MASK=ipadd] [METRIC=1..16] [MULTICAST={OFF SEND RECEIVE BOTH ON}] [POLICYFILTER={100..199 NONE}] [PRIORITYFILTER={200..299 NONE}] [PROXYARP={ON OFF}] [RIPMETRIC=1..16] [VJC={ON OFF}]	6-92
SET IP LOCAL [FILTER={filter-number NONE}] [IPADDRESS=ipadd] [POLICYFILTER={filter-number NONE}] [PRIORITYFILTER={filter-number NONE}]	6-94

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

SET IP NAMESERVER=ipadd	6-95
SET IP RIP INTERFACE=interface [CIRCUIT=miox-circuit] [IP=ipadd] [SEND={NONE RIP1 RIP2 COMPATIBLE}] [RECEIVE={NONE RIP1 RIP2 BOTH}] [DEMAND={NO YES}] [AUTH={NONE PASSWORD MD5}] [PASSWORD=password]	6-96
SET IP RIPTIMER [FLUSH=seconds] [HOLDDOWN=seconds] [INVALID=seconds] [UPDATE=seconds]	6-97
SET IP ROUTE FILTER=filter-id IP=ipadd MASK=ipadd ACTION={INCLUDE EXCLUDE} [DIRECTION={RECEIVE SEND BOTH}] [INTERFACE=interface] [NEXTHOP=ipadd] [POLICY=0..7] [PROTOCOL={ANY RIP STATIC INTERFACE}]	6-100
SET IP ROUTE TEMPLATE=name [NEXTHOP=ipadd] [CIRCUIT=miox-circuit] [METRIC=1..16] [METRIC1=1..16] [POLICY=0..7] [PREFERENCE=0..65535]	6-101
SET IP ROUTE=ipadd INTERFACE=interface MASK=ipadd NEXTHOP=ipadd [CIRCUIT=miox-circuit] [METRIC=1..16] [METRIC1=1..16] [POLICY=0..7] [PREFERENCE=0..65535]	6-98
SET IP SECONDARYNAMESERVER=ipadd	6-102
SET ISDN CALL=name [NUMBER=number] [PRECEDENCE={IN OUT}] [ALTNUMBER=number] [BUMPDELAY=0..100] [CALLBACK={ON OFF YES NO TRUE FALSE}] [CALLINGNUMBER=number] [CALLINGSUBADDRESS=calling-subaddress] [CBDELAY=0..100] [CHECKCLI={OFF PRESENT REQUIRED}] [CHECKSUB={OFF LOCAL REMOTE}] [CHECKUSER={OFF LOCAL REMOTE}] [CLLIST=0..99] [DIRECTION={IN OUT BOTH}] [DOV={ON OFF YES NO TRUE FALSE}] [HOLDUP=0..7200] [INANY={ON OFF YES NO TRUE FALSE}] [INTPREF={NONE interface}] [INTREQ={NONE interface}] [KEEPUP={ON OFF YES NO TRUE FALSE}] [LOGIN={ALL NONE CHAP PAP-RADIUS PAP-TACACS USER}] [OUTCLI={OFF CALLING INTERFACE NONUMBER}] [OUTSUB={OFF LOCAL REMOTE}] [OUTUSER={OFF LOCAL REMOTE}] [PASSWORD={NONE CLI CALLED SUB NAME USER}] [PPPTemplate=template] [PRIORITY=0..99] [RATE={56K 64K}] [REMOTECALL=name remote-number] [RN1=0..10] [RN2=0..5] [RT1=5..120] [RT2=300..1200] [SEARCHCLI={ON OFF YES NO TRUE FALSE CALLED 0..99}] [SEARCHSUB={OFF LOCAL REMOTE}] [SEARCHUSER={OFF LOCAL REMOTE}] [SUBADDRESS=number] [USER={ATTACH PPP}] [USERNAME={NONE CLI CALLED SUB NAME USER}]	4-75
SET ISDN DOMAINNAME=domain-name	4-80
SET ISDN LOG [PORT={0..23 NONE}] [LENGTH=0..100]	4-81
SET LAPD=interface [NASMODE={NORMAL MASTER SLAVE}] [NASMASTER=interface]	4-82
SET LAPD=interface {ATTACH=sap CONNECT=sap DATA=sap CES=ces ESTABLISH=sap CES=ces MDATA=sap CES=ces MUNIT=sap CES=ces RELEASE=sap CES=ces UNIT=sap CES=ces}	4-82
SET LAPD=interface DEBUG={OFF STATE PACKET}	4-82
SET LAPD=interface MODE={AUTOMATIC NONAUTOMATIC}	4-82
SET LAPD=interface SAP=sap {N200 N201 N202}=time...	4-82
SET LAPD=interface SAP=sap {T200 T201 T202 T203}=time...	4-82
SET LAPD=interface SAP=sap K=value	4-82
SET LOADER [DELAY=delay] [DESTINATION=FLASH] [FILE=filename] [METHOD={HTTP TFTP WEB WWW ZMODEM NONE}] [PORT=port] [PROXYPORT=proxyport] [SERVER=ipadd]	1-51
SET LOG OUTPUT={TEMPORARY output-id} [DESTINATION={MEMORY PORT ROUTER SYSLOG}] [FORMAT={FULL MSGONLY SUMMARY}] [MAXQUEUESEVERITY=severity] [MESSAGES=message-count] [PASSWORD={password NONE}] [PORT=port-number] [QUEUEONLY={YES NO}] [SECURE={YES NO}] [SERVER=ipadd] [ZONE={time-zone-name utc-offset}]	12-25
SET LOG OUTPUT={TEMPORARY output-id} FILTER=filter-id [ACTION={PROCESS IGNORE}] [ALL] [DATE=[op]dd-mmm-yyyy] [DEVICE=[op]device] [FILE=[op]filename] [MASK=ipadd] [MSGTEXT=[op]string] [MODULE=[op]module-id] [ORIGIN=ipadd] [REFERENCE=[op]string] [SEVERITY=[op]severity] [SOURCELINELINE=[op]line] [SUBTYPE=[op]subtype-id] [TIME=[op]hh:mm:ss] [TYPE=[op]type-id]	12-25
SET LOG RECEIVE={ipadd ANY} [ALLOW={YES NO}] [MASK=ipadd] [PASSWORD={password NONE}] [PROTOCOL={ALL BOTH NEW OLD SYSLOG}]	12-29
SET LOG UTCOFFSET={time-zone-name utc-offset}	12-30
SET MANAGER PORT={port-number NONE}	1-53
SET MIOX=x25t-interface [MINOPEN=10..360] [INACTIVE=10..360] [HOLDDOWN=10..360] [COLLISION=10..360] [FAILURE=10..360]	5-18
SET MIOX=x25t-interface CIRCUIT=circuit-name [{DTEADDRESS=dteaddress PVC=1..4095}] [CPAR=0..8] [ENCAP={IP NULL MULTIPLE}] [COMMENT=comment] [COMP={ON OFF}] [TCPCOMP={ON OFF}]	5-19

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

SET PASSWORD	1-54
SET PBX [CADENCE={BELL UNAV} VALUE=cadence-values] [COUNTRY={AUSTRALIA CHINA CUSTOM HOLLAND JAPAN KOREA NEWZEALAND UK USA}] [DATA=data-key] [DEBUG={ON OFF}] [DIAL={OVERLAP ENBLOC}] [DISCONNECT=5..10] [ENCODE={ULAW ALAW}] [FLASHHOOKMIN={2..4 OFF}] [INTERDIGIT={1..30 NONE}] [RESERVEBCHANNEL={ON OFF}]	14-20
SET PBX EXTENSION=extension-number [BCAP={SPEECH AUDIO}] [CALLINGNUMBER={calling-number OFF}] [COEFFICIENT={TH1 TH2 TH3 IM1 IM2 FRX FRR AX AR TG1 TG2} VALUE=coefficient-values] [COPY=extension-number] [GROUP=group-name] [HLC={DEFAULT FAX TELEPHONE}] [NAME=extension-name] [NOHLC={ACCEPT REJECT}] [NUMACCEPT={matching-number ALL NOTPRESENT OFF}] [PORT={pbx-interface NONE}] [SUBACCEPT={matching-subaddr ALL NOTPRESENT OFF}] [SUPPRESS={1..30 NONE}] [TERMINATE={0..30 NONE}]	14-21
SET PBX GROUP=group-name [EXTENSION=extension-number] [HUNT={SEARCH NONE}] [NUMACCEPT={matching-number ALL NOTPRESENT OFF}] [SUBACCEPT={matching-subaddr ALL NOTPRESENT OFF}]	14-24
SET PING [[IPADDRESS=]ipadd] [DELAY=seconds] [LENGTH=number] [NUMBER={number CONTINUOUS}] [PATTERN=hexnum] [SIPADDRESS=ipadd] [SCREENOUTPUT={YES NO}] [TIMEOUT=number] [TOS=number]	6-103
SET PORT[=port-number] [ATTENTION={BREAK ^P NONE}] [DATABITS={5 6 7 8}] [ECHO={ON OFF YES NO TRUE FALSE}] [FLOW={CHARACTER HARDWARE NONE}] [HISTORY=0..99] [INFLOW={CHARACTER HARDWARE NONE}] [MAXOQLEN=0..214783647] [NAME=name] [OUTFLOW={CHARACTER HARDWARE NONE}] [PAGE={4..99 OFF}] [PARITY={EVEN MARK NONE ODD SPACE}] [PROMPT={prompt DEFAULT OFF}] [SECURE={ON OFF YES NO TRUE FALSE}] [SPEED={AUTO 75 110 134.5 150 300 600 1200 1800 2000 2400 4800 9600 14400 14.4K 19200 19.2K 28800 28.8K 38400 38.4K 57600 57.6K 115200 115.2K}] [STOPBITS={1 2}] [TYPE={DUMB VT100}]	2-19
SET PPP TEMPLATE=template [AUTHENTICATION={CHAP EITHER PAP NONE}] [BAP={ON OFF}] [BAPMODE={CALL CALLBACK}] [CBDELAY=1..100] [CBMODE={ACCEPT OFF REQUEST}] [CBNUMBER=e164number] [CBOperation={E164NUMBER USERAUTH}] [COMPALGORITHM=STACLSZS] [COMPRESSION={ON OFF LINK}] [DEBUGMAXBYTES=16..256] [DESCRIPTION=description] [ECHO={ON OFF period}] [FRAGMENT={ON OFF}] [FRAGOVERHEAD=0..100] [IDLE={ON OFF time}] [INDATALIMIT={NONE 1..65535}] [IPPOOL={pool-name NONE}] [IPREQUEST={ON OFF}] [LOGIN=USER] [LQR={ON OFF time}] [MAGIC={ON OFF}] [MAXLINKS=1..64] [MULTILINK={ON OFF}] [NULLFRAGTIMER=time] [ONLINELIMIT={NONE 1..65535}] [OUTDALIMIT={NONE 1..65535}] [PASSWORD=password] [RESTART=time] [STACHECK={LCB SEQUENCE}] [TOTALDALIMIT={NONE 1..65535}] [USERNAME=username]	3-54
SET PPP[=ppp-interface] [OVER=physical-interface] [AUTHENTICATION={CHAP EITHER PAP NONE}] [AUTHMODE={IN OUT INOUT}] [BAP={ON OFF}] [BAPMODE={CALL CALLBACK}] [CBDELAY=1..100] [CBMODE={ACCEPT OFF REQUEST}] [CBNUMBER=e164number] [CBOperation={E164NUMBER USERAUTH}] [COMPALGORITHM=STACLSZS] [COMPRESSION={ON OFF LINK}] [CONFIGURE={value CONTINUOUS}] [DEBUGMAXBYTES=16..256] [DESCRIPTION=description] [DNSPRIMARY=ipadd] [DNSSECONDARY=ipadd] [DOWNRATE=0..100] [DOWNTIME=time] [ECHO={ON OFF period}] [FRAGMENT={ON OFF}] [FRAGOVERHEAD=0..100] [IDLE={ON OFF time}] [INDATALIMIT={NONE 1..65535}] [IPPOOL={pool-name NONE}] [IPREQUEST={ON OFF}] [LQR={ON OFF time}] [MAGIC={ON OFF}] [NULLFRAGTIMER=time] [NUMBER=number] [ONLINELIMIT={NONE 1..65535}] [OUTDALIMIT={NONE 1..65535}] [PASSWORD=password] [RESTART=time] [STACHECK={LCB SEQUENCE}] [TERMINATE={value CONTINUOUS}] [TOTALDALIMIT={NONE 1..65535}] [TYPE={DEMAND PRIMARY SECONDARY}] [UPRATE=0..100] [UPTIME=time] [USERNAME=username] [WINSPRIMARY=ipadd] [WINSSECONDARY=ipadd]	3-49
SET Q931=interface [DOVNUMBER=number] [INTID={hex-string}] [NONUM={ACCEPT REJECT}] [NOSUB={ACCEPT REJECT}] [NUM1=number] [NUM2=number] [PROFILE={5ESS AUS CHINA DMS-100 ETSI JAPAN KOREA NI1 NZ}] [RATE={56K 64K}] [SPID1=spid] [SPID2=spid] [SUB1=subaddress] [SUB2=subaddress] [timer={OFF time}]	4-84
SET SCRIPT=filename LINE=line [AFTER=line] [BEFORE=line] [TEXT=text]	13-10
SET SNMP COMMUNITY=name [ACCESS={READ WRITE}] [OPEN={ON OFF YES NO TRUE FALSE}]	16-19
SET SYSTEM CONTACT=contact-name	1-54
SET SYSTEM LOCATION=location	1-55
SET SYSTEM NAME=name	1-55

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
SET SYSTEM TERRITORY={AUSTRALIA CHINA EUROPE JAPAN KOREA NEWZEALAND USA}																								1-56	
SET TELNET [TERMTYPE=termstring] [INSERTNULL={ON OFF}]																								7-6	
SET TRACE [[IPADDRESS=]ipadd] [MAXTTL=number] [MINTTL=number] [NUMBER=number] [PORT=port-number] [SCREENOUTPUT={YES NO}] [SOURCE=ipadd] [TIMEOUT=number] [TOS=number]																								6-104	
SET TRIGGER=trigger-id [CPU=value [DIRECTION={UP DOWN ANY}]] [INTERFACE=interface EVENT={UP DOWN FAIL ANY} [CIRCUIT=miox-circuit] [CP={BCP CCP PCP LCP}] [MEMORY=value [DIRECTION={UP DOWN ANY}]] [PERIODIC=minutes] [REBOOT={RESTART CRASH ALL}] [TIME=hh:mm] [DATE=date] [DAYS=day-list] [AFTER=hh:mm] [BEFORE=hh:mm] [SCRIPT=filename...] [NAME=name] [REPEAT={YES NO ONCE FOREVER count}] [STATE={ENABLED DISABLED}] [TEST={YES NO ON OFF}]																								10-13	
SET TTY [HISTORY=0..99] [PAGE=4..99] [PROMPT={string-15 DEFAULT OFF}] [TYPE={DUMB VT100}]																								7-7	
SET USER [LOGINFAIL=1..10] [LOCKOUTPD=0..30000] [MANPWDFAIL=1..5] [SECUREDELAY=10..600] [MINPWDLEN=1..23]																								1-57	
SET USER=login-name [CBNUMBER=e164number] [DESCRIPTION=description] [PASSWORD=password] [PRIVILEGE={USER MANAGER}] [TELNET={YES NO}]																								1-57	
SET X25T CPAR=call-index [MAXDATA={128 256 512 1024}] [RMAXDATA={128 256 512 1024}] [RWINDOW=1..127] [TMAXDATA={128 256 512 1024}] [TWINDOW=1..127] [USERDATA=hex-string] [WINDOW=1..127]																								5-22	
SET X25T=x25-interface [MAXACTIVE=0..4095] [MODULUS={8 128}] [T20=1..360] [T21=1..360] [T22=1..360] [T23=1..360] [T24={1..360 OFF}] [T27={1..360 OFF}] [MINRECALL=1..360] [R20=0..65535] [R22=0..65535] [R23=0..65535] [R27=0..65535] [NPVC=0..4095] [DEFCPAR=0..8] [DTEADDRESS=dteaddress] [LIC=0..4095] [HIC=0..4095] [LTC=0..4095] [HTC=0..4095] [LOC=0..4095] [HOC=0..4095] [ROLE={DYNAMIC DCE DTE}]																								5-21	
SHOW BOOTP RELAY																								6-105	
SHOW BRI[=instance] CONFIGURATION																								4-86	
SHOW BRI[=instance] COUNTERS[={INTERFACE BRI}]																								4-88	
SHOW BRI[=instance] CTEST																								4-94	
SHOW BRI[=instance] DEBUG																								4-95	
SHOW BRI[=instance] STATE																								4-96	
SHOW BRI[=instance] TEST																								4-100	
SHOW BUFFER [SCAN=[address [QUEUEPOINTERS]]]																								1-59	
SHOW CONFIG [DYNAMIC[=module-id]]																								1-61	
SHOW CPU																								1-63	
SHOW DEBUG																								1-63	
SHOW DHCP																								15-22	
SHOW DHCP CLIENT [RANGE=name]																								15-23	
SHOW DHCP POLICY[=name]																								15-24	
SHOW DHCP RANGE[=name]																								15-25	
SHOW ENCO																								8-8	
SHOW ENCO CHANNEL[=channel [COUNTERS]]																								8-9	
SHOW ENCO COUNTERS={JOBPROCESSING STAC USER UTIL}																								8-13	
SHOW ETH=n CONFIGURATION																								2-22	
SHOW ETH[=n] COUNTERS[={COLLISION DIAGNOSTIC DOT3STAT INTERFACE}]																								2-23	
SHOW ETH[=n] MACADDRESS																								2-29	
SHOW ETH[=n] RECEIVE																								2-29	
SHOW EXCEPTION																								1-64	
SHOW FFILE[=file-identifier] [CHECK]																								1-65	
SHOW FILE[=filename]																								1-67	
SHOW FIREWALL																								17-25	
SHOW FIREWALL POLICY=name [COUNTERS] [SUMMARY]																								17-26	
SHOW FIREWALL SESSION[=session-number] [POLICY=name] [COUNTERS] [PORT={port-port service-name}] [PROTOCOL={protocol ALL EGP ICMP OSPF TCP UDP}] [SUMMARY]																								17-32	
SHOW FLASH [FFS]																								1-67	

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

SHOW FLASH PHYSICAL	1-69
SHOW HTTP CLIENT	1-69
SHOW HTTP DEBUG	1-70
SHOW HTTP SERVER	1-72
SHOW HTTP SESSION	1-71
SHOW INSTALL	1-74
SHOW INTERFACE[={ifIndex interface}] [COUNTERS]	2-30
SHOW IP	6-106
SHOW IP ARP	6-108
SHOW IP COUNTER[={ALL ICMP INTERFACE IP MULTICAST RIP ROUTES UDP}]	6-109
SHOW IP DEBUG[=1..40]	6-116
SHOW IP FILTER[=filter-number]	6-117
SHOW IP HELPER [COUNTER]	6-119
SHOW IP HOST	6-120
SHOW IP INTERFACE[=interface] [COUNTER[=MULTICAST]]	6-121
SHOW IP POOL[=pool-name] [IP=ipadd[-ipadd]] [SUMMARY]	6-124
SHOW IP RIP [INTERFACE=interface] [CIRCUIT=miox-circuit] [IP=ipadd]	6-126
SHOW IP RIP COUNTER[={DETAIL SUMMARY}] [INTERFACE=interface] [CIRCUIT=miox-circuit] [IP=ipadd]	6-128
SHOW IP RIPTIMER	6-127
SHOW IP ROUTE FILTER	6-134
SHOW IP ROUTE TEMPLATE[=name]	6-135
SHOW IP ROUTE[=ipadd] [{GENERAL CACHE COUNT}]	6-130
SHOW IP TRUSTED	6-136
SHOW IP UDP	6-136
SHOW ISDN CALL[={acnum name}]	4-103
SHOW ISDN CLILIST[=0..99]	4-107
SHOW ISDN DOMAINNAME	4-108
SHOW ISDN LOG	4-109
SHOW LAPD[=interface]	4-110
SHOW LAPD[=interface] COUNT	4-112
SHOW LAPD[=interface] STATE	4-114
SHOW LOADER	1-75
SHOW LOG COUNTERS	12-38
SHOW LOG OUTPUT[={TEMPORARY output-id}] [{FILTER=filter-id FULL}]	12-40
SHOW LOG QUEUE[=output-id]	12-44
SHOW LOG RECEIVE[=ipadd] [MASK=ipadd]	12-45
SHOW LOG STATUS	12-46
SHOW LOG[=output-id] [DATE=[op]dd-mmm-yyyy] [DEVICE=[op]device] [FILE=[op]filename] [FULL] [MASK=ipadd] [MODULE=[op]module-id] [MSGONLY] [MSGTEXT=[op]string] [ORIGIN=ipadd] [REFERENCE=[op]string] [REVERSE[=count]] [SEVERITY=[op]severity] [SOURCELIN=[op]line] [SUBTYPE=[op]subtype-id] [TAIL[=count]] [TIME=[op]hh:mm:ss] [TYPE=[op]type-id] [ZONE={time-zone-name utc-offset}]	12-31
SHOW MANAGER PORT	1-77
SHOW MIOX[=x25t-interface]	5-23
SHOW MIOX[=x25t-interface] CIRCUIT[=circuit-name] [COUNTER ENCAP]	5-26
SHOW MIOX[=x25t-interface] COUNT	5-24
SHOW PBX	14-25
SHOW PBX CALL	14-26
SHOW PBX EXTENSION[=extension-number]	14-27
SHOW PBX GROUP[=group-name]	14-29

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

SHOW PING	6-137
SHOW PORT[=port-number ALL] [{COUNTERS[={DIAGNOSTIC INTERFACE RS232}]] HISTORY SUMMARY}]	2-33
SHOW PPP NAMESERVER	3-80
SHOW PPP TEMPLATE[= <i>template</i>] [DEBUG]	3-80
SHOW PPP[=ppp-interface]	3-58
SHOW PPP[=ppp-interface] CONFIG	3-59
SHOW PPP[=ppp-interface] COUNT[={INTERFACE LCP MULTILINK NCP}]	3-65
SHOW PPP[=ppp-interface] DEBUG	3-75
SHOW PPP[=ppp-interface] IDLETIMER	3-76
SHOW PPP[= <i>ppp-interface</i>] LIMITS	3-77
SHOW PPP[=ppp-interface] MULTILINK	3-78
SHOW PPP[= <i>ppp-interface</i>] TXSTATUS	3-84
SHOW Q931[=interface] [CALL[=q931-call]]	4-114
SHOW Q931[=interface] SPID	4-117
SHOW SCRIPT[=filename]	13-11
SHOW SNMP	16-20
SHOW SNMP COMMUNITY=name	16-22
SHOW STARTUP	1-77
SHOW SYSTEM	1-78
SHOW TCP[=tcb]	6-139
SHOW TDM GROUP[=groupname] [INTERFACE=interface]	11-6
SHOW TEST [INTERFACE] [COUNTERS]	9-10
SHOW TIME	1-80
SHOW TRACE	6-143
SHOW TRIGGER[=trigger-id] [{COUNTERS FULL STATUS SUMMARY}]	10-16
SHOW TTY[=tty-number ALL] [{SUMMARY DEFAULT}]	7-8
SHOW USER[=login-name] [CONFIGURATION]	1-80
SHOW X25T CPAR[=call-index]	5-34
SHOW X25T[=x25-interface] [{CIRCUIT COUNT}]	5-30
STOP PING	6-145
STOP TRACE	6-145
TELNET {ipadd host}	7-11
TRACE [{IPADDRESS=ipadd} [MAXTTL=number] [MINTTL=number] [NUMBER=number] [PORT=port-number] [SCREENOUTPUT={YES NO}] [SOURCE=ipadd] [TIMEOUT=number] [TOS=number]	6-146
UPLOAD [FILE=filename] [SERVER=ipadd] [PORT=port] [METHOD=ZMODEM]	1-82
WAIT delay	13-12

Chapter 1

Operation

Introduction	1-3
The Command Processor	1-3
User Privilege Levels	1-3
Entering Commands	1-4
Aliases	1-5
Online Help	1-5
Storing and Retrieving Configuration Information	1-6
User Authentication Facility	1-7
The User Authentication Database	1-8
Asynchronous Port Security	1-11
Telnetting from the Router	1-11
Counters	1-11
Semipermanent Manager Port	1-12
Remote Management	1-12
Monitoring and Fault Diagnosis	1-13
Event Logging	1-13
Restarts	1-13
CPU Utilisation	1-13
Memory	1-13
FLASH Memory	1-14
Physical Characteristics	1-14
The File Subsystem	1-15
File Naming Conventions	1-15
Using Wildcards to Specify Groups of Files	1-16
Working With Files	1-16
FLASH File System	1-16
Working with FFS Files	1-17
Compaction	1-17
FFS Messages	1-18
The Built-in Editor	1-18
HTTP Client and Server	1-19
Resolving Uniform Resource Locators (URLs)	1-20
Software Releases	1-20
Releases	1-20
Router Startup Operations	1-21
Downloading Releases into the Router	1-22
Install Information	1-23
Examples	1-24
Command Reference	1-26
ACTIVATE FLASH COMPACTION	1-26
ADD ALIAS	1-27
ADD USER	1-28
CLEAR FLASH TOTALLY	1-29

CREATE CONFIG	1-29
CREATE FFILE	1-30
DELETE ALIAS	1-31
DELETE FFILE	1-31
DELETE FILE	1-32
DELETE INSTALL	1-32
DELETE USER	1-33
DISABLE HTTP DEBUG	1-33
DISABLE HTTP SERVER	1-34
DISABLE USER	1-34
DUMP	1-35
EDIT	1-36
ENABLE HTTP DEBUG	1-39
ENABLE HTTP SERVER	1-39
ENABLE USER	1-40
HELP	1-40
LOAD	1-41
LOGIN	1-43
LOGOFF	1-44
MODIFY	1-44
PURGE USER	1-45
RENAME	1-45
RESET HTTP SERVER	1-46
RESET LOADER	1-46
RESET USER	1-47
RESTART	1-48
SET CONFIG	1-48
SET HELP	1-49
SET INSTALL	1-49
SET LOADER	1-50
SET MANAGER PORT	1-52
SET PASSWORD	1-53
SET SYSTEM CONTACT	1-53
SET SYSTEM LOCATION	1-54
SET SYSTEM NAME	1-54
SET SYSTEM TERRITORY	1-55
SET TIME	1-55
SET USER	1-56
SHOW ALIAS	1-57
SHOW BUFFER	1-58
SHOW CONFIG	1-60
SHOW CPU	1-62
SHOW DEBUG	1-62
SHOW EXCEPTION	1-63
SHOW FFILE	1-64
SHOW FILE	1-66
SHOW FLASH	1-66
SHOW FLASH PHYSICAL	1-68
SHOW HTTP CLIENT	1-68
SHOW HTTP DEBUG	1-69
SHOW HTTP SESSION	1-70
SHOW HTTP SERVER	1-71
SHOW INSTALL	1-72
SHOW LOADER	1-73
SHOW MANAGER PORT	1-75
SHOW STARTUP	1-75
SHOW SYSTEM	1-76
SHOW TIME	1-78
SHOW USER	1-78
UPLOAD	1-80

Introduction

This section describes the functions and commands available on the router to support day-to-day operational and network management activities.

The commands described in this section fall into six functional groups:

- The command processor and router configuration.
- The User Authentication Facility.
- Monitoring and fault diagnosis of the router and the network.
- Managing FLASH memory and the FLASH File System (FFS).
- Downloading software releases and enhancements.

The Command Processor

The router is controlled and monitored with a set of commands which can be entered from a terminal connected to one of the asynchronous ports, or by using Telnet to connect to the router.

A user accessing the router from a terminal connected to an asynchronous port in secure mode, or via a Telnet connection, must enter a login name and password to gain access to the command prompt (see “*User Authentication Facility*” on page 1-7).

The command processor supports two levels of privilege, USER and MANAGER. USER and MANAGER privilege can be distinguished by the prompt displayed by the command processor when it is ready to receive commands. A USER level prompt looks like:

>

while a MANAGER prompt looks like:

Manager >

If the router's system name has been defined with the command:

```
SET SYSTEM NAME=name
```

then the system name is included in the prompt. The MANAGER level prompt for a router with the system name `ho.noname.com` looks like:

```
Manager ho.noname.com>
```

User Privilege Levels

The commands that can be executed by a user depend on the user's privilege level and whether the router is operating in normal or security mode:

The USER level has access to a very limited subset of commands, regardless of whether the router is operating in normal or security mode. USER level commands only affect the user's own session or asynchronous port. USER privilege applies to a user who has not logged in (i.e. is using a terminal connected to an asynchronous port that is **not** in secure mode), or a user who has logged in to a username with USER privilege.

The MANAGER level has access to the full set of commands. MANAGER privilege can be gained in one of two ways:

- Using the command:

LOGIN

from any port or Telnet session to login under a login name that has MANAGER privilege. The command prompts for a login name and password. The password is case-sensitive and must be entered exactly as defined. If the password is entered correctly, the port or Telnet connection gains MANAGER privilege and the prompt changes to the MANAGER level prompt. This is the usual method of gaining MANAGER privilege, especially when managing remote routers.

- Using the command:

SET MANAGER PORT

to set a particular port as a semipermanent MANAGER port. Any terminal connected to the specified port will have MANAGER privilege. The SET MANAGER PORT command on page 1-53 is a MANAGER level command and can only be entered from a port or a Telnet session that already has MANAGER privilege. Only one port at a time can be defined as manager port.

To return to USER mode, use the command:

LOGOFF



Normally, the prompt changes when the user's privilege level changes from USER to MANAGER or vice versa. The prompt will not change if commands are being entered from a terminal connected to a physical port and the port's PROMPT parameter has been changed to a user-defined string with the SET PORT command on page 2-19 of Chapter 2, Interfaces.

Entering Commands

The router supports command line editing and recall. The functions available are:

- Move the cursor backwards and forwards in the command line, using the cursor keys.
- Move the cursor to either end of the command line with a single keystroke.
- Insert and delete characters.
- Clear the command line.
- Toggle between insert and overstrike editing modes.
- Recall, edit and execute previous commands.
- Move backwards and forwards through a history of previous commands.
- Display a command history and select a command from the list.
- Clear the command history.
- Recall the most recent command matching a partially entered command.

Table 1-1 on page 1-5 lists the functions and the terminal keys or key combinations used to access these functions.

Table 1-1: Command line editing functions and keystrokes.

Function	VT100 Terminal	Dumb terminal
Move cursor within command line	←, →	<i>Not available</i>
Delete character to left of cursor	[Delete] or [Backspace]	[Delete] or [Backspace]
Toggle between insert/overstrike	[Ctrl/O]	<i>Not available</i>
Clear command line	[Ctrl/U]	[Ctrl/U]
Recall previous command	↑ or [Ctrl/B]	[Ctrl/B]
Recall next command	↓ or [Ctrl/F]	[Ctrl/F]
Display command history	[Ctrl/C] or SHOW PORT HISTORY	[Ctrl/C] or SHOW PORT HISTORY
Clear command history	RESET PORT HISTORY	RESET PORT HISTORY
Recall matching command	[Tab] or [Ctrl/I]	[Tab] or [Ctrl/I]

The router assumes that the width of the terminal screen is 80 characters, and performs command line wrapping at the 80th column regardless of the setting of the terminal. The cursor does not need to be at the end of the line for the command to be executed. The default editing mode is insert mode. Characters are inserted at the cursor position and any characters to the right of the cursor are pushed to the right to make room. In overstrike mode, characters are inserted at the cursor position and replace any existing characters.

Aliases

The command line interface supports aliases. An alias is a short name for an often-used longer character sequence. When the user presses [Enter] to execute the command line, the command processor first checks the command line for aliases and substitutes the replacement text. The command line is then parsed and processed normally. Alias substitution is not recursive—the command line is scanned only once for aliases.

Aliases are created and destroyed using the commands:

```
ADD ALIAS=name STRING=substitution
DELETE ALIAS=name
```

A list of all the aliases defined on the router and their replacement strings can be displayed using the command:

```
SHOW ALIAS
```

Online Help

Online help is available for all router commands. Typing a question mark “?” at the end of a partially completed command displays a list of the parameters that may follow the current command line, with the minimum abbreviations in uppercase letters (Figure 1-1 on page 1-6). The current command line is then re-displayed, ready for further input.

Figure 1-1: Using the question mark character ("?") to display help for the current command.

```
Manager > ADD ?
```

```
Options : BOOTp IP ISDN LAPD LOG MIOX PPP SScript SNmp TRGger
        USer X25T TDM
```

A multilingual, language-independent online help facility provides more detailed help information via the command:

```
HELP [topic]
```

If a topic is not specified, a list of available topics is displayed. The HELP command on page 1-41 displays information from the system help file stored in FLASH memory. The help file uses a simple mark-up language to identify topics, access level (USER or MANAGER) and help text. Both standard ASCII and Unicode character encodings are supported. Alternate help files can be uploaded and stored in FLASH, then activated using the command:

```
SET HELP=helpfile
```

The current help file can be displayed with the command:

```
SHOW SYSTEM
```

The help file is easily modified, for example to provide detailed site-specific support information. The mark-up language specification and preprocessor program are available from your distributor or reseller.

Storing and Retrieving Configuration Information

At boot the router executes the commands in the boot script to configure the router. The default boot script is called `boot.cfg`, but an alternative script file can be defined as the boot script using the command:

```
SET CONFIG=filename
```

Subsequent commands entered from the command line or executed from a script affect only the dynamic configuration in memory, which is not retained over a power cycle. Changes are not automatically stored in nonvolatile memory. When the router is restarted the configuration will be restored to that defined by the boot script, or if the router was restarted using the RESTART command on page 1-49, any script specified in the RESTART command.

To ensure that any configuration changes made after boot are retained across a restart or power cycle, the modified configuration must be saved as a script file, using the command:

```
CREATE CONFIG=filename
```



The CREATE CONFIG command on page 1-30 writes the MD5 digest, not the clear text, of passwords in commands to the configuration file. When a configuration script is executed the command processor can determine whether the password value is clear text or an MD5 digest.

If the file name specified is `boot.cfg`, or the file is set as the boot script using the SET CONFIG command on page 1-49, the modified configuration will automatically be restored after a restart or power cycle. If another name is

specified, the configuration can be restored after a restart or power cycle using the command:

```
ACTIVATE SCRIPT=filename
```

User Authentication Facility

The *User Authentication Facility* (UAF) controls access to the router's command prompt, asynchronous services and dialup services via a login name and password. A user will be prompted to enter a login name and password when:

- The user attempts to access the router's command prompt via a terminal connected directly to an asynchronous port set to SECURE mode.
- The user attempts to access the router's command prompt via a Telnet connection.
- The user enters the LOGIN command on page 1-44.

The UAF prompts the user for a login name and password (Figure 1-2 on page 1-7). The user must enter appropriate responses, pressing [Return] after each response. Characters entered at the password prompt are not echoed to the screen, for security reasons.

Figure 1-2: A typical login session for user BRUCE on router CMD.

```
CMD login: bruce
password:

CMD >
```

If the user enters an invalid login name or password, the sequence is repeated a set number of times. If a valid login name and password has still not been entered the terminal or Telnet session is *locked out* for a period of time. During this period the password prompt is withheld, preventing the user from logging in or entering commands. The manager can specify the number of login attempts allowed and the length of the lockout period.



The password prompt is displayed regardless of whether or not a password is required for the login name entered by the user. This makes it more difficult for an intruder to discover valid login name/password combinations.

The UAF supports an internal database called the *User Authentication Database*.

The UAF queries the User Authentication Database. If the supplied login name and password does not match an entry in the User Authentication Database, the login is rejected.

The User Authentication Database

The User Authentication Database stores information about the users who are permitted to have access to the router's command prompt, asynchronous services and dialup services. Users are identified by a login name. Each login name has an associated record in the database which specifies:

- The password that the user must enter to login to the router.
- The privilege level for the user: USER, or MANAGER.
- Whether or not the user is permitted to use the TELNET command on page 7-11 of *Chapter 7, Terminal Server*, or to connect to a Telnet service from a Telnet session.
- A callback number for use with the PPP callback facility.

Adding Entries to the User Authentication Database

When the router is started up for the first time one account is created automatically. This account has the login name MANAGER, the password "friend", and MANAGER privilege. This account can not be deleted, although the password may be changed.



The manager should change the password of the MANAGER account at the earliest opportunity. Leaving the MANAGER account with the default password is a security risk, as the account name and default password are well documented.

Additional users can be added to the User Authentication Database using the command:

```
ADD USER=login-name PASSWORD=password [CBNUMBER=e164number]  
      [DESCRIPTION=description] [PRIVILEGE={USER|MANAGER}]  
      [TELNET={YES|NO}]
```

The number of entries in the database is limited only by the amount of memory available. Only the login name and password must be specified. The default privilege level is USER. Other information about a user that may be specified includes a description for the entry (e.g. the user's full name), the privilege level, and whether or not the user is permitted to use the TELNET command on page 7-11 of *Chapter 7, Terminal Server*. The callback number is only required if the user is to make a PPP callback request with user authentication. See *Chapter 3, Point-to-Point Protocol (PPP)* for more information.

Modifying Entries in the User Authentication Database

An entry in the database can be modified with the command:

```
SET USER=login-name [PASSWORD=password] [CBNUMBER=e164number]  
      [DESCRIPTION=description] [PRIVILEGE={USER|MANAGER}]  
      [TELNET={YES|NO}]
```

An entry in the database can be deleted using the command:

```
DELETE USER=login-name
```

All entries in the database, except the MANAGER account, can be deleted with the command:

```
PURGE USER
```


The contents of the database can be displayed with the command:

```
SHOW USER [=login-name]
```

Passwords

All users, including managers, should take care in selecting passwords. Tools exist that enable hackers to guess or test many combinations of login names and passwords easily. The UAF provides some protection against such attacks by allowing the manager to set the number of consecutive login failures allowed and a lockout period when the limit is exceeded.

However, the best protection against password discovery is to select a good password, and keep it secret. When choosing a password:

- Do make it six or more characters in length. The UAF enforces a minimum password length, which can be changed by the manager. The default is six characters.
- Do include both alphabetic (a–z) and numeric (0–9) characters.
- Do include both uppercase and lowercase characters. The passwords stored by the router are case-sensitive, so “bgz4kal” and “Bgz4Kal” are different.
- Do avoid words found in a dictionary, unless combined with other random alphabetic and numeric characters.
- **Do not** use the login name, or the word “password” as the password.
- **Do not** use your name, your mothers name, your spouses name, your pets name, or the name of your favourite cologne, actor, food or song.
- **Do not** use your birth date, street number or telephone number.
- **Do not** write down your password anywhere.

A manager can alter the password for any user with the command:

```
SET USER=username PASSWORD=password
```

This may be necessary if the user has forgotten the password. A log message is generated whenever the password for a manager account is changed.

A user who is logged in can change their own password using the command:

```
SET PASSWORD
```

which prompts for the old password, the new password and confirmation of the new password. The new password and the confirmation must be identical for the change to take affect. This reduces the chances of a typing error causing the password to be different from what the user intended.

Database Security

A manager session that is left unattended is a severe security risk. In particular, the User Authentication Database can be modified from a manager session. To reduce the risk of unauthorised activity, a subset of manager commands (Table 1-2 on page 1-10), called the *security commands*, have a *security timer*. When one of the security commands is entered from a manager session, the security timer is started. Each time a security command is entered the timer is restarted. If a security command is entered after the timer has expired, the manager is prompted to re-enter the password correctly before the command will be actioned. If the password is not entered correctly the password prompt

will be repeated a set number of times, and if the correct password is still not entered a log message is generated and the session is logged off.

The security timer enables a manager to make successive additions and modifications to the database at one time without having to re-enter the password for every command.



The security timer does not provide a foolproof security mechanism. Managers should always attempt to log out of a manager session before leaving a terminal unattended.

Table 1-2: Secure commands controlled by the security timer.

Command	Description
ADD USER	Adds a user to the User Authentication Database.
DELETE USER	Deletes a user from the User Authentication Database.
PURGE USER	Deletes all users except MANAGER from the User Authentication Database.
SET MANAGER PORT	Assigns a port semipermanent MANAGER privilege.
SET USER	Modifies a user record in the User Authentication Database.

Logging In and Logging Out

A user will automatically be prompted to enter a login name and password when attempting to access the router via Telnet or a terminal connected to an asynchronous port set to SECURE mode.

There are other occasions when a user may wish to login manually. A user on a terminal connected to an asynchronous port that is not in SECURE mode may wish to login in order to use facilities that are only available to logged in users, such as the TELNET command on page 7-11 of *Chapter 7, Terminal Server*. A user who is already logged in may wish to temporarily login as another user in order to acquire different rights, such as MANAGER privilege.

To log in to the router manually, use one of the commands:

```
LOGIN
LOGON
LOGI
```

which are synonyms. To log out of a session, use one of the commands:

```
LOGOFF
LOGOUT
LO
```

which are synonyms.



If a user Telnets to the router but does not attempt to login within one minute, the router automatically times out the session and terminates the Telnet connection.

Recovering Lost Passwords

If a user forgets their password, the password can be reset from an account with MANAGER privilege, using the command:

```
SET USER=login-name PASSWORD=password
```

Passwords for accounts with MANAGER privilege can be reset with the same command, provided the manager can login to at least one account with MANAGER privilege.

Asynchronous Port Security

Asynchronous ports may be set to SECURE mode, using the command:

```
SET PORT SECURE=ON
```

See *Chapter 2, Interfaces* for a detailed description of the SET PORT command on page 2-19 of *Chapter 2, Interfaces*. By default, all asynchronous ports are set to SECURE mode. Telnet sessions are always in SECURE mode. A user accessing the router via a terminal connected to an asynchronous port in SECURE mode, or via Telnet, must login before the router will accept any other commands. When a user Telnets to a router the login and password prompts are always displayed. The password prompt is displayed even if the login name does not match an entry in the User Authentication Database, to make it more difficult for an intruder to discover a valid login name. When a login name and password is entered that does not match an entry in the database, the login sequence is repeated. If successive login failures occur, the login prompt is withheld for a specified *lockout period*. This makes it much more difficult for an intruder to randomly try login names and passwords hoping to gain entry. A log message is generated when the number of retries for a connection is exceeded and the lockout period is instigated. Telnet logins from an offending IP address are also locked out for this period once the permitted number of failures is exceeded. The number of login attempts permitted and the length of the lockout period can be configured with the command:

```
SET USER [LOGINFAIL=1..10] [LOCKOUTPD=0..30000]
```

Telnetting from the Router

Users logged into the router via a terminal attached to an asynchronous port can use the TELNET command on page 7-11 of *Chapter 7, Terminal Server* to Telnet to remote hosts.

Users logged into an account with the TELNET attribute set to "ON" can use the TELNET command on page 7-11 of *Chapter 7, Terminal Server* to Telnet to remote hosts.

Counters

A number of counters record activity associated with the User Authentication Database. Counters relating to specific users in the database can be displayed with the command:

```
SHOW USER[=login-name]
```

Global counters and configuration parameters can be displayed with the command:

```
SHOW USER CONFIGURATION
```

All counters are stored in nonvolatile storage so that they are retained across router reboots and power cycles.

The counters for a specific user can be reset to zero using the command:

```
RESET USER=login-name
```

The counters for all users, the global counters, or all counters can be reset to zero with the command:

```
RESET USER COUNTERS={USER|GLOBAL|ALL}
```

Semipermanent Manager Port

It is sometimes desirable to have an asynchronous port that has MANAGER privilege after a router reboot, without a manager having to log on. An asynchronous port can be set to default to MANAGER privilege using the command:

```
SET MANAGER PORT=port-number
```

Only one port may be a semipermanent manager port. By default, no semipermanent manager port is defined. This command is defined as one of the security commands (see “*Database Security*” on page 1-9).

When the router boots with a semipermanent manager port configured, the MANAGER account is automatically logged in to the port. The port has full MANAGER privilege and there is no restriction on Telneting from the port. The security timer is reset so that the first time a security command is entered the user will be challenged for the password for the MANAGER account.

Remote Management

Managing remote routers is as easy as managing the local router to which the terminal is connected. From a terminal connected to any port (with either USER or MANAGER privilege), use the command:

```
TELNET ipadd
```

to Telnet to the remote router, specifying the remote router’s IP address. If the connection is successful a login prompt from the remote router is displayed. Login using a login name that has been defined with MANAGER privilege (such as the default MANAGER login name), and enter the password.

To return to the local router, use the command:

```
LOGOFF
```

to terminate the connection. For more information about using Telnet, see *Chapter 7, Terminal Server*.

Monitoring and Fault Diagnosis

Event Logging

The router responds to certain significant events by generating an event log message. Each router maintains a local event log of the most recent log messages. To view the log, use the command:

```
SHOW LOG
```

The logging facility provides a powerful, flexible and easily configurable tool for monitoring network activity and selecting and displaying the results. User-defined output definitions can filter, prioritise and output log messages to RAM, an asynchronous port, another router, or a syslog server. See *Chapter 12, Logging Facility* for a detailed description of the logging facility.

Restarts

Some changes to configuration parameters require the router to be restarted for the changes to take affect. The router is restarted with the command:

```
RESTART {REBOOT|ROUTER} [CONFIG={filename|NONE}]
```

If the router encounters a fatal error condition from which it can not recover, it automatically performs a restart. The reason for the restart may be determined by examining the router's exception list, with the command:

```
SHOW EXCEPTION
```

The conditions that the router encountered when it last restarted, such as the amount of RAM and the state of the battery-backed RAM, can be viewed with the command:

```
SHOW STARTUP
```

A complete snapshot of the state of the router prior to the last fatal condition can be displayed with the command:

```
SHOW DEBUG
```

CPU Utilisation

The CPU utilisation over the last second, ten seconds, one minute or since the router last restarted can be displayed with the command:

```
SHOW CPU
```

Memory

The state of the router's buffer pool can be examined with the command:

```
SHOW BUFFER
```

If the pool of free buffers drops below a critical threshold, the router progressively disables processes, resulting in a loss of functionality. This problem can potentially arise when a fast source sends enormous amounts of data to a slow destination or down a slow link. However, the cause is more

likely to be a problem with the router itself. The problem can be corrected in the short term by restarting the router, but it should be reported to your supplier.

The contents of memory can be examined with the command:

DUMP

and modified with the command:

MODIFY



The DUMP command on page 1-35 and the MODIFY command on page 1-45 are provided as diagnostic tools and should not be needed for normal operation of the router. Inappropriate use of these commands may cause a malfunction of the router, resulting in the loss of network services.

FLASH Memory

FLASH memory is a nonvolatile, reusable memory device that allows large volumes of data (up to 8MB) to be stored in the router. The primary function of FLASH memory in the router is to store multiple software releases, simplifying the servicing and maintenance requirements of the router. Releases can be remotely loaded into FLASH memory from any router port using the Loader Module. Multiple software releases can be loaded and then individually selected for use at runtime by the Install Module. Comprehensive management features are provided to examine the state of the FLASH memory and to view or modify the contents.

To enable FLASH memory to support applications other than just software releases it is structured like a disk subsystem with files which can be created, deleted, read and written by any router module. Files can also be manipulated directly using the command line interface. This allows FLASH to be used to store any type of data, including releases, configurations and logs.

Physical Characteristics

FLASH memory is a special type of nonvolatile memory which can be erased and reprogrammed many times in-situ. FLASH memory has advantages over other types of nonvolatile memory in that it has a very large storage capacity and it does not require power from a battery to retain stored data. The main limitations of FLASH memory are that it has a fixed erase block size, so individual bytes can not be changed without first clearing a whole block of data, and a limit on the number of erase cycles that can be performed. However, the erase limit is very high, typically at least 100000 cycles, which would allow three erases per day for 100 years before the limit was exceeded.

In the router, FLASH memory can be installed directly onto the system board during manufacture, or subsequently as FLASH SIMM sticks mounted on the 80-pin SIMM connector.



The FLASH SIMM sticks used are specially designed for the router and must be obtained from your distributor or reseller.

The presence and amount of FLASH memory installed is displayed using the command:

```
SHOW SYSTEM
```

More detailed information about the FLASH memory can be displayed using the command:

```
SHOW FLASH PHYSICAL
```

The File Subsystem

The file subsystem provides a consistent file-based interface to all physical memory devices on the router used for data storage, including FLASH memory. The file subsystem allows data, such as code releases and configuration scripts to be stored on the router in a file structure and manipulated in the same way with the same commands, regardless of where the file is physically stored.

File Naming Conventions

The file subsystem provides a flat file system—directories are not supported. Files are uniquely identified by a file name of the form:

```
[device:]filename.ext
```

where:

- *device* specifies the physical memory device on which the file is stored, and must be FLASH. If *device* is specified, it must be separated from the rest of the file name by a colon (":"). If *device* is not specified, the default is FLASH.
- *filename* is a descriptive name for the file, and may be one to eight characters in length. Valid characters are lowercase letters (a–z), uppercase letters (A–Z), digits (0–9) and the hyphen character (-).
- *ext* is a file name extension, one to three characters in length. Valid characters are lowercase letters (a–z), uppercase letters (A–Z), digits (0–9) and the hyphen character (-). The extension is used by the router to determine the data type of the file and how to use the file (Table 1-3 on page 1-15). If *ext* is specified, it must be separated from the *filename* portion by a period (".")

Table 1-3: File extensions and file types.

Extension	File type/function
CFG	Configuration or boot script
HLP	Help file
LOG	Log file
MDS	Modem script
REL	Software release
REZ	Compressed release
SCP	Script
TXT	Generic text file

The following are examples of valid file names:

<code>flash:config.scp</code>	A script file.
<code>flash:28-72.rel</code>	Software Release 7.2.

The following are examples of illegal file names:

<code>flash:/sys/head_o.cfg</code>	"/" is not a valid delimiter character, and directories are not supported.
<code>flash:headoffice.cfg</code>	The filename is too long. A maximum of eight characters is allowed.

Using Wildcards to Specify Groups of Files

The asterisk character ("*") may be used as a wildcard character in some commands to identify a groups of files to be processed by the command. A wildcard must replace an entire field of the file name — *device*, *filename* or *ext*. A wildcard can not be combined with other characters. The following are examples of valid wildcard expressions:

```
flash:*. *
*:*.rel
```

The following is not a valid wildcard expression:

```
flash:28*.rel
```

Working With Files

To display a directory of the files stored on the router, use the command:

```
SHOW FILE
```

To limit the display to certain files, use the command:

```
SHOW FILE=filename
```

filename may contain wildcard characters. Files can be permanently deleted using the command:

```
DELETE FILE=filename
```

filename may contain wildcard characters. Files can be created using the router's built-in editor, using the command:

```
EDIT [filename]
```

or by downloading the file via HTTP, TFTP or ZMODEM, using the command:

```
LOAD FILE=filename
```

FLASH File System

The FLASH File System (FFS) provides additional functionality on top of that provided by the file subsystem, to manage the peculiarities of FLASH technologies. The additional functionality of the FFS includes:

- Header and data integrity is ensured with a checksum mechanism.
- All FLASH processes can recover from a power cycle without data loss.

- Automatic recovery of deleted file space by the compaction process.

Information about the state of the FFS can be displayed using the command:

```
SHOW FLASH
```

Working with FFS Files

FFS files can be managed like any other file on the router, using the standard file subsystem commands:

```
EDIT [filename]
DELETE FILE=filename
LOAD FILE=filename
SHOW FILE [=filename]
```

In addition, the following commands can be used to manage files stored in FLASH memory. To display a directory of the files stored in FLASH memory, use the command:

```
SHOW FFILE [CHECK]
```

If CHECK is specified then the file data checksum is also verified. This is included as an option because it can take some time to complete a check on large files. A file data check is also carried out each time a file is read by the system.

A FLASH file can be deleted with the command:

```
DELETE FFILE=filename
```

Wildcards are allowed in the *filename* and *ext* fields of the file name, but are not allowed in the device field. The file is marked as deleted but the space occupied by the file is not freed until the next compaction process.

The FLASH memory can be completely erased using the command:

```
CLEAR FLASH TOTALLY
```



This command totally erases all stored FLASH information and reformats the FLASH file structure.

Compaction

FLASH memory has a granular erase structure which requires data to be erased in large blocks rather than as individual bytes. To allow files to be mapped onto this structure the FFS keeps track of the status of each file — whether it is being written, is complete or is deleted. When the total amount of FLASH memory used for deleted files reaches a preset limit a compaction process is initiated. Compaction searches through the FLASH memory copying good files to a new location. As soon as all the good files within an erase block have been copied the block is cleared. This results in any deleted files present in the block being cleared, freeing up space for new files. If there is a large amount of FLASH memory in use then the compaction process can take several seconds to complete. However, FLASH memory operations continue to operate without being affected by the compaction process.



Compaction can be interrupted by a router restart or power cycle without effect — the router will simply continue the compaction process when router operation resumes.

Compaction can also be manually initiated using the command:

```
ACTIVATE FLASH COMPACTION
```

FFS Messages

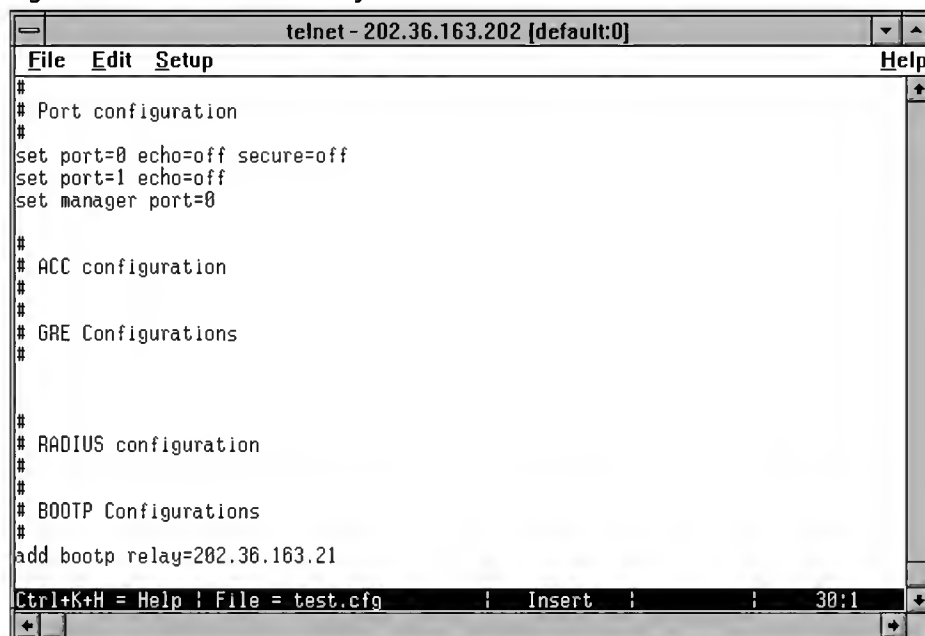
Some FFS processes generate messages in the system log (displayed with the SHOW LOG command on page 12-31 of *Chapter 12, Logging Facility*) which include FFS message codes. See “FLASH File System Message Codes” on page B-4 of *Appendix B, Reference Tables* for a complete list of the possible codes and their meanings.

The Built-in Editor

The router has a built-in full-screen text editor for editing ASCII text files stored on the router file subsystem.

The editor uses VT100 command sequences and should only be used with a VT100-compatible terminal, terminal emulation program or Telnet client. The VT100 screen only supports 24 lines, unlike a PC. Lines 1–23 are used to display the text of the file being edited, and line 24 is used as the status bar and command line (Figure 1-3 on page 1-18). The status bar displays the current file name, line and column position in the file and the editing mode (overstrike or insert). When additional command information is required, such as a file name or search text, then a prompt is displayed in the status bar.

Figure 1-3: The editor screen layout.



The editor is invoked with the command:

```
EDIT [filename]
```

The file name is optional as a file can be loaded, or a new file can be created from within the editor itself. The editor is currently limited to editing one file at

a time. To overcome this limitation use the cut and paste facility to transfer text between files.



Before starting the editor make sure your terminal, terminal emulation program or Telnet client is 100% compatible with a VT100 terminal.

Help can be obtained at any time while in the editor by pressing [Ctrl/K,H]; that is, holding down the Ctrl key and pressing in turn the K key then the H key.

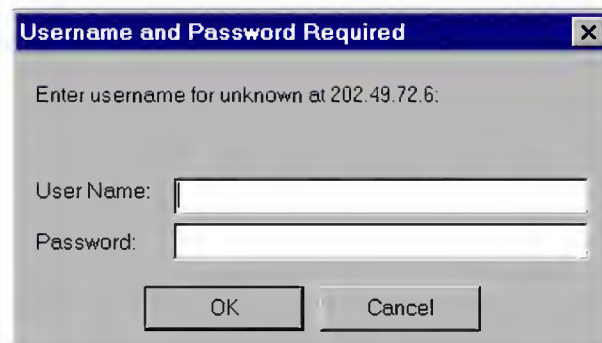
HTTP Client and Server

The router has a built-in HTTP client and server. The HTTP server is compatible with any HTTP/1.1-compliant browser and allows the router to serve HTML pages out of FLASH memory to a remote web browser. It also allows users to login into the router and dynamically configure the router via the Configuration Wizard, a web-based GUI (*Graphical User Interface*). The HTTP server is enabled by default. To disable the HTTP server, or to enable the HTTP server after it has been disabled, use the commands:

```
DISABLE HTTP SERVER
ENABLE HTTP SERVER
```

When a user attempts to access the router via a web browser, the HTTP server will request authentication from the browser. The browser will prompt the user for a username and password (Figure 1-4 on page 1-19).

Figure 1-4: Logging in to the router from a web browser.



The username and password entered by the user must match a user defined in the User Authentication Database (see “*The User Authentication Database*” on page 1-8).

All GET, configure and monitor requests, and authorisation failures are logged to the Logging Facility (see *Chapter 12, Logging Facility*). Debugging can be enabled or disabled using the commands:

```
ENABLE HTTP DEBUG={ALL|AUTH|MSG|SESSION}
DISABLE HTTP DEBUG={ALL|AUTH|MSG|SESSION}
```

Debug messages display authorisation attempts, HTTP GET and POST requests and responses, and TCP state changes. The currently enabled debugging options can be displayed using the command:

```
SHOW HTTP DEBUG
```

The command:

```
RESET HTTP SERVER
```

restarts the HTTP server, disables debugging and clears all counters.

To display the current status of the HTTP server, use the command:

```
SHOW HTTP SERVER
```

To display information about the currently active sessions on the HTTP server, use the command:

```
SHOW HTTP SESSION
```

The HTTP client enables the router to act as a browser by sending HTTP GET or POST requests to another HTTP server. The HTTP client is used by the Configuration Wizard to download updates from a support web site. To display the current status of the HTTP client, use the command:

```
SHOW HTTP CLIENT
```

Resolving Uniform Resource Locators (URLs)

When the HTTP server receives a request for a URL, it uses the following procedure to resolve the URL:

- If the URL matches the name of a file stored in the router's FLASH memory, the file will be loaded and sent to the browser.
- If the URL does not match the name of a file stored in FLASH, the HTTP server searches a list of dynamically generated HTML pages for a match. If a match is found the page is generated and sent to the browser.
- If the URL does not match the name of a file stored in FLASH or the name of a dynamically generated HTML page, the HTTP server will return the HTML error 404, indicating the URL could not be found.

Software Releases

Software releases are identified by a number of the format *<major>.<minor>.<interim>*. The release with interim release number is "0" is known as the "base release". For example, Software Release 1.9.0 is the base release of 1.9, and Software Release 1.9.1 is the first interim release of 1.9.

Releases

A software release contains a copy of the system software that executes on the router. Releases are given numbers that look like "1.9.0". In this case the *major* release number is "1", the *minor* release number is "9" and the *interim* release number is "0". Releases are stored in FLASH memory.

A standard release is a single file with a name of the form:

`1-rrr.REL`

where `rrr` is the release number (e.g. 191 for Software Release 1.9.1).

There are two methods of providing compressed releases, depending on the release number of the base EPROMs in the router. For Software Release 7.4 and later, compressed releases are supported by the base EPROMs and the file required for a compressed release is:

`mm-rrr.REZ`

A number of releases can be stored in the router at once. The router contains INSTALL information that specifies which release is to be loaded at boot. This information may be changed at any time.



A software release is specific to a particular router series. It is not possible to run a release on any router series other than that for which the release was made. The same router release will, however, run on all models in the same series. If an attempt is made to load the wrong software release into the router the boot process will fail.

Router Startup Operations

When the router boots, the following sequence of operations is performed:

1. Perform startup self tests.
2. Perform the install override option.
3. Load the EPROM release as the INSTALL boot.
4. Inspect and check INSTALL information.
5. Load the required EPROM or FLASH release as the main boot.
6. Start the router.
7. Execute the boot script, if one has been configured.

If a terminal is connected to port 0, a series of status and progress messages, similar to those shown in Figure 1-5 on page 1-21, are displayed during the startup process.

Figure 1-5: Router startup messages.

```
INFO: Self tests beginning.
INFO: RAM test beginning.
PASS: RAM test, 4096k bytes found.
INFO: BBR tests beginning.
PASS: BBR test, 128k bytes found.
PASS: BBR test. Battery OK.
INFO: Self tests complete
INFO: Downloading router software.
Force EPROM download (Y) ?
INFO: Initial download succeeded
INFO: Executing configuration script <boot.cfg>
INFO: Router startup complete

Manager >
```

The startup self tests check the basic operation of the router. A router that passes these tests should be able to at least proceed far enough to perform the load of the EPROM release and to start operating.

The install override option is designed to allow a mandatory router boot from the EPROM release. The message:

```
Force EPROM download (Y)?
```

is displayed on the terminal connected to port 0 and the router pauses. If a key is not pressed within a few seconds, the startup process will continue and all steps in the sequence will be executed. If the [Y], [S] or [N] key on the terminal is pressed immediately after the message is displayed, the router startup process can be altered (Table 1-4 on page 1-22).

Table 1-4: Router startup sequence keystrokes.

Pressing key...	Forces the router to...
Y	Load the EPROM release, and skip straight to step 6.
S	Start with the default configuration. Any boot script is ignored.
[Ctrl/D]	Enter diagnostics mode.

The EPROM release is always loaded first when starting the router. This release contains all the code required to obtain and check the INSTALL information. This first boot is known as the INSTALL boot. The INSTALL information is inspected and the router set up to perform another load. Even if the actual release required is the EPROM release, another load is always performed.

The router startup occurs immediately after the install override option, or after the INSTALL information check. This performs a full startup of router software and initiates the normal operation of the router.

Finally, if a boot script has been defined, the script is executed.

Downloading Releases into the Router

The LOADER module is responsible for loading and storing releases and other files into FLASH. The LOADER module uses the *Trivial File Transfer Protocol* (TFTP), *Hypertext Transfer Protocol* (HTTP) or ZMODEM over an asynchronous port, to retrieve files from a network host. The FFS module is used to create, write and destroy files.

The loader can be configured with the command:

```
SET LOADER
```

This command sets default values for the name of the file to load, the network host to load it from, and the memory location in which to store the file. These default values can be overridden when the load actually takes place. A time delay between initiating a load and the start of the load can also be configured.

The configuration of the LOADER module can be displayed with the command:

```
SHOW LOADER
```

This shows the default configuration for the LOADER module as well as the status of any current file transfer.

To actually initiate a load, use the command:

```
LOAD
```

This command will use either the default values for the LOADER module or the values specified on the command line. The command:

```
SHOW LOADER
```

displays the progress of the load. The current load can be stopped at any time using the command:

```
RESET LOADER
```

leaving the LOADER module ready to load again. Only one file can be loaded at a time. Another load can not be initiated while loading is in progress.

Once the release file has been loaded, its presence can be checked either with the command:

```
SHOW FILE
```

A release file can be removed with the command:

```
DELETE FILE
```

Files to be loaded by the LOADER module must be resident on a TFTP server accessible via the network, or accessible via the ZMODEM protocol over an asynchronous port. Release files are ASCII files, and consist of a header followed by a sequence of Motorola S-records containing the actual code for the release. The header has a standard format, which provides information about the release to the router.



The header in the release file should not be altered. At best, this will cause the file load or install to fail, at worst the router could be put into a state where it will not boot correctly until field service action is taken.

Install Information

The INSTALL module is responsible for maintaining install information and loading the correct install at boot. An *install* is a record identifying a release. Three installs are maintained by the INSTALL module, *temporary*, *preferred* and *default*.

The default install is the install of last resort. The release for the default install can not be changed by the manager and is always the EPROM release.

The temporary and preferred installs are completely configurable. The release may be EPROM or a release stored in FLASH.

The three different installs are required to handle the following situations:

- A default install is required to handle the case when only the EPROM release is present.
- A temporary install is required to allow a release to be loaded once only, in case it causes a router crash.
- A preferred install is required because the default install can not be anything other than the EPROM.

The install information is inspected in a strict order. The temporary install is inspected first. If this install information is present, the temporary install is

loaded. At the same time, the temporary install information is deleted. This ensures that if the router reboots immediately as the result of a fatal condition caused by the temporary install, the temporary install will not be loaded a second time.

If there is no temporary install defined, or the install information is invalid, the preferred install is inspected. If present, this install is loaded. The preferred install information is never deleted.

If neither temporary nor preferred installs are present, the default install is used. The default install will always be present in the router, because if, for some reason, it is not, the INSTALL module will restore it.



The preferred install should not be set up with an untested release. It is advisable to install new releases as the temporary install, and when the router boots correctly, to then set up the preferred install with the new release.

To change the install information in the router, use the command:

```
SET INSTALL
```

To delete a particular install (except the default install) use the command:

```
DELETE INSTALL
```

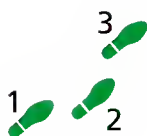
To display the current install information, including which install is currently running in the router, and how the install information was checked at the last reboot, use the command:

```
SHOW INSTALL
```

Examples

Installing a Standard Release using TFTP

This example assumes that the router is correctly configured to allow TFTP to function. This means that IP has been configured and the router is able to communicate with the designated TFTP server. The TFTP server is assumed to be functioning correctly and the release files are assumed to be present in the server's TFTP directory. The router has no release files, and is running the EPROM Software Release 7.6.0. The IP address of the server is 172.16.1.1. The name of the release file being loaded is 8-761.rel.



To install a standard release:

1. Configure the loader.

The LOADER module is set up with defaults to make the process of downloading files in future simpler. All release files in this router will be stored in FLASH.

```
SET LOADER SERVER=172.16.1.1 DEST=FLASH
```

2. Download the release file to the router.

The release file is downloaded to the router with the command:

```
LOAD FILE=8-761.REL
```


The process of downloading a release file can take some time, even if the router and the TFTP server are connected by high speed links. An indicative time for downloading a release over Ethernet is 5 to 10 minutes. The progress of the download can be monitored with the command:

```
SHOW LOAD
```

When the download has completed, the presence of the file in FLASH can be displayed with the command:

```
SHOW FILE
```

This shows the file 8-761.rel is present.

3. Test the release.

The release can now be tested, using the command:

```
SET INSTALL=TEMPORARY RELEASE=8-761.REL
```

The install information can be checked with the command:

```
SHOW INSTALL
```

The router is then rebooted, and the install is checked again. This display should indicate, in the install history, that the temporary install was loaded.

4. Make the release the default (permanent) release.

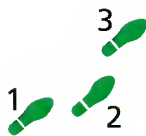
If the router operates correctly with the new release, the release may be made permanent with the command:

```
SET INSTALL=PREFERRED RELEASE=8-761.REL
```

Every time the router reboots from now on, the new release will be loaded from FLASH.

Installing a Compressed Release

This example is identical to the previous example, except that a compressed release is installed.



To install a compressed release:

1. Configure the loader.

The LOADER module is set up with defaults to make the process of downloading files in future simpler.

```
SET LOADER SERVER=172.16.1.1 DEST=FLASH
```

2. Download the release files to the router.

The compressed release files are downloaded to the router with the commands:

```
LOAD FILE=8-761.REZ
```

The process of downloading a release file can take some time, even if the router and the TFTP server are connected by high speed links. An indicative time for downloading a release over Ethernet is 5 to 10 minutes. The progress of the download can be monitored with the command:

```
SHOW LOAD
```

When the download has completed, the presence of the files in FLASH can be displayed with the command:

```
SHOW FILE
```

This shows the file 8-761.rez is present.

3. Test the release.

The release can now be tested, using the command:

```
SET INSTALL=TEMPORARY RELEASE=8-761.REZ
```

The install information can be checked with the command:

```
SHOW INSTALL
```

The router is then rebooted, and the install is checked again. This display should indicate, in the install history, that the temporary install was loaded.

4. Make the release the default (permanent) release.

If the router operates correctly with the new release, the release may be made permanent with the command:

```
SET INSTALL=PREFERRED RELEASE=8-761.REZ
```

Every time the router reboots from now on, the new release will be loaded from FLASH.

Command Reference

This section describes the commands available on the router to support day-to-day operational and management activities.

See “Conventions” on page xxxv of *Preface* for details of the conventions used to describe command syntax. See *Appendix A, Messages* for a complete list of messages and their meanings.

ACTIVATE FLASH COMPACTION

Syntax ACTIVATE FLASH COMPACTION

Description This command activates the FLASH compaction process. Compaction is the process of cleaning up garbage (deleted files) by searching through FLASH memory copying valid files to a new block and erasing the old blocks. The compaction process normally occurs automatically when the amount of garbage reaches a preset limit, so manual compaction is not required for normal operation. This command can be used to recover garbage space before the automatic compaction threshold is reached.

Compaction is required because the FLASH memory has a granular erase structure which requires data to be erased in large blocks rather than as individual bytes. To allow files to be mapped onto this structure the FFS keeps track of the status of each file — whether it is being written, is complete or is deleted. When the total amount of FLASH memory used for deleted files reaches a preset limit a compaction process is initiated. Compaction searches through the FLASH memory copying good files to a new location. As soon as all the good files within an erase block have been copied the block is cleared.

This results in any deleted files present in the block being cleared, freeing up space for new files. If there is a large amount of FLASH memory in use then the compaction process can take several seconds to complete. However, FLASH memory operations continue to operate without being affected by the compaction process.



Compaction can be interrupted by a router restart or power down without effect—the router will simply continue the compaction process when router operation resumes.

While compaction is underway the command:

```
SHOW FLASH
```

will indicate an FFS global operation of “compacting”. When compaction is complete the global operation will return to “none”.

See Also `SHOW FLASH`

ADD ALIAS

Syntax `ADD ALIAS=name STRING=substitution`

Where:

- *name* is a character string 1 to 132 characters in length. It may contain any printable character. If *name* contains spaces it must be enclosed in double quotes. It is case-sensitive.
- *substitution* is a character string 1 to 132 characters in length. It may contain any printable character. If *substitution* contains spaces it must be enclosed in double quotes. It is case-sensitive.

Description This command adds a new alias for a longer character sequence. When the user presses [Enter] to execute the command line, the command processor first checks the command line for aliases and substitutes the replacement text. The command line is then parsed and processed normally. Alias substitution is not recursive—the command line is scanned only once for aliases. An alias may represent either part of a command, or a complete command.

The ALIAS parameter specifies the name of the alias. This is the text that the user enters on the command line.

The STRING parameter specifies the substitution string. When the command processor parses the command line, all occurrences of the alias are replaced by this string.

Examples To create an alias “df” that expands to “delete file=1-190.rez”, use the command:

```
add alias=df string="delete file=1-190.rez"
```

Thereafter, the following commands are equivalent:

```
df
del file=1-190.rez
```

See Also ADD ALIAS
DELETE ALIAS

ADD USER

Syntax `ADD USER=login-name PASSWORD=password
[CBNUMBER=e164number] [DESCRIPTION=description]
[PRIVILEGE={USER|MANAGER}] [TELNET={YES|NO}]`

where:

- *login-name* is a character string, 1 to 32 characters in length. Valid characters are uppercase letters (A–Z), lowercase letters (a–z), and decimal digits (0–9). The string may not contain spaces.
- *password* is a character string, 1 to 32 characters in length. Valid characters are any printable character. If the string contains spaces it must be enclosed in double quotes.

- *e164number* is the phone number to dial when performing callback. It may contain digits (0–9) and should be a valid phone number as described in CCITT standard E.164.
- *description* is a character string, 1 to 23 characters in length. Valid characters are any printable character. If the string contains spaces it must be enclosed in double quotes.

Description This command adds a user to the User Authentication Database. The **USER** parameter specifies the login name for the user. It is case insensitive.

The **PASSWORD** parameter specifies the password for the user. The password is case sensitive. It is intended that the **PASSWORD** parameter be used to set an initial password for the user and that the user will change it to some string known only to the user, using the **SET PASSWORD** command on page 1-54. A password set with the **SET PASSWORD** command may contain any printing character. A configurable minimum password length is enforced. The default is 6 characters.

The **CBNUMBER** parameter specifies the ISDN phone number to use when making a call back to a remote user using the PPP callback facility.

The **DESCRIPTION** parameter specifies a descriptive text for the entry, such as the full name and location of the user. This string may contain any printing character and the case is preserved in output.

The **PRIVILEGE** parameter specifies the privilege level for the user. The default is **USER**. A user with **USER** privilege has access to only a limited subset of commands, generally commands that only affect the user's own session or asynchronous port. A user with **MANAGER** privilege has access to the complete router command set.

The **TELNET** parameter specifies whether or not the user is permitted to use the **TELNET** command on page 7-11 of *Chapter 7, Terminal Server* to Telnet to another host when logged in via Telnet.

Examples To add a user with the login name "BRUCE", the password "sbf4d4Q" and **MANAGER** privilege, use the command:

```
ADD USER=BRUCE DESCRIPTION="Bruce Wilson" PASSWORD=sbf4d4Q
PRIVILEGE=MANAGER
```

See Also DELETE USER
DISABLE USER
ENABLE USER
PURGE USER
RESET USER
SET USER
SHOW USER

CLEAR FLASH TOTALLY

Syntax CLEAR FLASH TOTALLY

Description This command completely clears the FLASH memory to an erased state. Clearing the FLASH memory is not required for normal operation. This command intended as a troubleshooting tool to allow the FLASH file system to be returned to a known state.



This command will destroy all existing files and reformat the FLASH memory. Files cannot be salvaged after the FLASH memory has been erased.

While the erasure is under way the SHOW FLASH command on page 1-67 will indicate that the FFS global operation is in the “erasing” state. When the erasure is complete a message is displayed and the global operation returns to “none”.



The operation of erasing FLASH may take up to a minute to complete.

See Also SHOW FLASH

CREATE CONFIG

Syntax CREATE CONFIG=*filename*

where:

- *filename* is a file name of the form `device:filename.ext`. Valid characters are the lowercase letters (a–z), digits (0–9) and the hyphen character (-). Wildcards are not allowed.

Description This command creates a script file containing the commands required to recreate the current dynamic configuration of the router.

The CONFIG parameter specifies the name of the script or configuration file to create. The file extension must be “`scp`” or “`cfg`”. If the file already exists, it is replaced. If the file does not exist it is created.



The CREATE CONFIG command on page 1-30 writes the MD5 digest, not the clear text, of passwords in commands to the configuration file. When a configuration script is executed the command processor can determine whether the password value is clear text or an MD5 digest.



The configuration of a specific software module can not be saved with this command. To save the configuration of a specific software module, use the SHOW CONFIG command on page 1-61 to display the configuration, capture the output and save it to a file.

Examples To save the current dynamic configuration as the default boot script `boot.cfg`, use the command:

```
CREATE CONFIG=BOOT.CFG
```

See Also RESTART
SET CONFIG
SHOW CONFIG

CREATE FFILE

Syntax `CREATE FFILE=filename {DATA=bytes|ADDRESS=address
LENGTH=length}`

where:

- *filename* is a file name of the form `device:filename.ext`. Valid characters are the lowercase letters (a–z), digits (0–9) and the hyphen character (-). Wildcards are not allowed.
- *bytes* is a comma-separated list of up to 80 byte values, expressed as hexadecimal numbers.
- *address* is a memory address, expressed as a hexadecimal number.
- *length* is a length in bytes, expressed as a hexadecimal number.

Description This command is used to create an FFS file. It is intended primarily for testing purposes, and should not be required during normal operation. There are two variants of the command. The first variant is used to create small files, and the DATA parameter specifies the bytes to be written to the file. The second variant is used to create larger files by copying data from elsewhere in the router's memory space. The ADDRESS parameter specifies the source address in memory and the LENGTH parameter specifies the number of bytes to copy to the new file, starting at the specified address.



Care must be taken when using this command to avoid creating an invalid file which a module will then try to use. If a module recognises the file name it may try to use the file, with unpredictable results if the file contents are not in the expected format.



Do not use this command unless specifically instructed to do so by your distributor or reseller.

Examples To create a file called FLASH:TINY.FIL containing the five bytes 0xCD, 0x20, 0x5, 0x7F and 0x28, use the command:

```
CREATE FFILE=FLASH:TINY.FIL DATA=CD,20,5,7F,28
```

To create a file called FLASH:BIG.FIL, of length 0xC0000, from the contents of memory starting at address 0x00, use the command:

```
CREATE FFILE=FLASH:BIG.FIL ADDRESS=0 LENGTH=C0000
```

See Also DELETE FFILE
SHOW FFILE

DELETE ALIAS

Syntax `DELETE ALIAS=name`

Where:

- *name* is a character string 1 to 132 characters in length. It may contain any printable character. If *name* contains spaces it must be enclosed in double quotes. It is case-sensitive.

Description This command deletes an existing alias. Occurrences of the alias string in the command line will no longer be expanded to the substitution text.

The ALIAS parameter specifies the name of the alias to be deleted.

Example To delete an alias with name "ii", use the command:

```
DELETE ALIAS=ii
```

See Also ADD ALIAS
SHOW ALIAS

DELETE FFILE

Syntax DELETE FFILE=*filename*

where:

- *filename* is a file identifier of the form `device:filename.ext`. Valid characters are the lowercase letters (a–z), digits (0–9) and the hyphen character (-). Wildcards are allowed in the name and extension elements.

Description This command deletes an FFS file. Wildcards are allowed in the name and type elements of the file identifier.



Caution must be taken when deleting files, such as releases and configurations, since they contain information which is vital to the intended operation of the router.

Examples To delete the file FLASH:28-68.REL, use the command:

```
DELETE FFILE=FLASH:28-68.rel
```

To delete all files in FLASH, use the command:

```
DELETE FFILE=FLASH:*. *
```

See Also CREATE FFILE
SHOW FFILE

DELETE FILE

Syntax DELETE FILE=*filename*

where:

- *filename* is a file identifier of the form [device:]name.ext. Valid characters are the lowercase letters (a–z), digits (0–9) and the hyphen character (-). Wildcards are allowed in the name and extension elements.

Description This command deletes the specified file or files. Wildcards are allowed in the name and extension elements of the file identifier.



Caution must be taken when deleting files, such as releases and configurations, since they contain information which is vital to the intended operation of the router.

Examples To delete the release file 28-72.REL, use the command:

```
DELETE FILE=28-72.REL
```

See Also RENAME
SHOW FILE

DELETE INSTALL

Syntax DELETE INSTALL={TEMPORARY|PREFERRED|DEFAULT}

Description This command deletes the specified install from the install information.

The INSTALL module is responsible for maintaining install information and loading the correct install at boot. An *install* is a record identifying a release. Three installs are maintained by the INSTALL module, *temporary*, *preferred* and *default*.

The default install is the install of last resort. The release for the default install can not be changed by the manager and is always the EPROM release.

The temporary and preferred installs are completely configurable. The release may be EPROM or a release stored in FFS.

Examples To delete the temporary install, use the command:

```
DELETE INSTALL=TEMPORARY
```

See Also SET INSTALL
SHOW INSTALL

DELETE USER

Syntax DELETE USER=*login-name*

where:

- *login-name* is a character string, 1 to 32 characters in length. Valid characters are uppercase letters (A–Z), lowercase letters (a–z), and decimal digits (0–9). The string may not contain spaces.

Description This command deletes a user from the User Authentication Database. The USER parameter specifies the login name for the user. It is case insensitive.

See Also ADD USER
DISABLE USER
ENABLE USER
PURGE USER
RESET USER
SET USER
SHOW USER

DISABLE HTTP DEBUG

Syntax DISABLE HTTP DEBUG={ALL|AUTH|MSG|SESSION}

Description This command disables HTTP server debugging. Debug output is sent to the terminal session or Telnet connection from which the command was entered.

The DEBUG parameter specifies the type of debugging to be disabled. If AUTH is specified, debugging of authentication attempts is disabled. If MSG is specified, debugging of HTTP GET and SET requests and responses, is disabled. If SESSION is specified, debugging of TCP state changes and session activity is disabled. If ALL is specified, all debugging is disabled. Debugging is disabled by default.

Examples To disable HTTP server debugging, use the command:

```
DISABLE HTTP DEBUG
```

See Also DISABLE HTTP SERVER
ENABLE HTTP DEBUG
ENABLE HTTP SERVER
RESET HTTP SERVER
SHOW HTTP CLIENT
SHOW HTTP DEBUG
SHOW HTTP SERVER
SHOW HTTP SESSION

DISABLE HTTP SERVER

Syntax DISABLE HTTP SERVER

Description This command disables the HTTP server. The HTTP server serves HTML pages out of the router's FLASH memory to a web browser, and allows users to login into the router and dynamically configure the router via a web-based GUI (*Graphical User Interface*). The server is enabled by default.

Examples To disable the HTTP server, use the command:

```
DISABLE HTTP SERVER
```

See Also DISABLE HTTP DEBUG
ENABLE HTTP DEBUG
ENABLE HTTP SERVER
RESET HTTP SERVER
SHOW HTTP CLIENT
SHOW HTTP DEBUG
SHOW HTTP SERVER
SHOW HTTP SESSION

DISABLE USER

Syntax `DISABLE USER=login-name`

where:

- *login-name* is a character string, 1 to 32 characters in length. Valid characters are uppercase letters (A–Z), lowercase letters (a–z), and decimal digits (0–9). The string may not contain spaces.

Description This command temporarily disables a user login name. The login name must be currently enabled. The USER parameter specifies the login name for the user. It is case insensitive. Login attempts using the login name will be ignored.

See Also ADD USER
DELETE USER
ENABLE USER
PURGE USER
RESET USER
SET USER
SHOW USER

DUMP

Syntax `DUMP [ADDR=address] [LEN=length] [SIZE={BYTE|LONG|WORD}]
[SPACE={SD|SP|UD|UP|UR}]`

where:

- *address* is the first address (in hexadecimal) to be dumped.
- *length* is the number of bytes (in hexadecimal) to dump.

Description This command displays the contents of the router's memory. The block of memory to be displayed is specified by the parameters ADDR, LEN and SPACE. The parameter SPACE specifies which of the possible CPU address spaces is to be dumped (Table 1-5 on page 1-36)

Table 1-5: Router CPU address spaces.

SPACE value	CPU address space
UD	User Data
UP	User Program
UR	User Reserved
SD	Supervisor Data
SP	Supervisor Program

The SIZE parameter specifies whether the data should be displayed grouped as BYTES, LONGWORDS or WORDs. Note that LEN is always in bytes, regardless of the value of SIZE.

If the LEN, SIZE or SPACE parameters are omitted then they default to the value they had at the previous invocation of the command. If the ADDR parameter is omitted it will increment to dump the block of memory immediately following the block dumped by the previous invocation. If the ADDR parameter is given without a value (e.g. just the string ADDR or ADDR=) then it will dump the block of memory previously dumped.



It is possible to use this command to dump I/O devices. This may interrupt the operation of the router. The DUMP command is provided mainly as a diagnostic tool. It should not be needed for normal operation of the router.

A typical display is shown in Figure 1-6 on page 1-36. The left-hand column shows the address of the data in each row. The next eight columns give the data starting at the address for the next 16 bytes. The right-most column is an ASCII representation of the data in the row, with non-printing characters represented by a dot.

Figure 1-6: Example output from the DUMP command.

```

00000000 0001 667c 0001 667c 0000 b424 0001 667c      ..f|..f|...$.f|
00000010 0001 667c 0001 667c 0001 667c 0001 667c      ..f|..f|..f|..f|
00000020 0001 667c 0001 667c 0001 667c 0001 667c      ..f|..f|..f|..f|
00000030 0001 667c 0001 667c 0001 667c 0001 667c      ..f|..f|..f|..f|
00000040 0001 667c 0001 667c 0001 667c 0001 667c      ..f|..f|..f|..f|
00000050 0001 667c 0001 667c 0001 667c 0001 667c      ..f|..f|..f|..f|
00000060 0001 66d4 0001 6b14 0001 667c 0001 667c      ..f|..f...k...f|
00000070 0001 667c 0001 1308 0001 6aa4 0001 66c8      ..f|.....j...f.
00000080 0001 667c 0001 667c 0001 667c 0001 667c      ..f|..f|..f|..f|
00000090 0001 667c 0001 667c 0001 667c 0001 667c      ..f|..f|..f|..f|
000000a0 0001 667c 0001 667c 0001 667c 0001 667c      ..f|..f|..f|..f|
000000b0 0001 667c 0001 667c 0001 667c 0001 667c      ..f|..f|..f|..f|
000000c0 0001 667c 0001 667c 0001 667c 0001 667c      ..f|..f|..f|..f|
000000d0 0001 667c 0001 667c 0001 667c 0001 667c      ..f|..f|..f|..f|
000000e0 0001 667c 0001 667c 0001 667c 0001 667c      ..f|..f|..f|..f|
000000f0 0001 667c 0001 667c 0001 667c 0001 667c      ..f|..f|..f|..f|

```

Examples The command used to produce the output shown above was:

```
DUMP ADDR=0 LEN=100 SIZE=WORD SPACE=SD
```

See Also MODIFY

EDIT

Syntax EDIT [*filename*]

where:

- *filename* is a file name of the form `device:filename.ext`. Valid characters are the lowercase letters (a–z), digits (0–9) and the hyphen character (-). Wildcards are not allowed.

Description This command invokes the router's built-in full-screen text editor to edit an ASCII text file. If a filename is specified then the editor will load the file if it exists on the system. If the device field is not specified, the default is `FLASH`.

The editor uses VT100 command sequences (Table 1-6 on page 1-38) and should only be used with a VT100-compatible terminal, terminal emulation program or Telnet client.

Table 1-6: Editor functions and keystrokes.

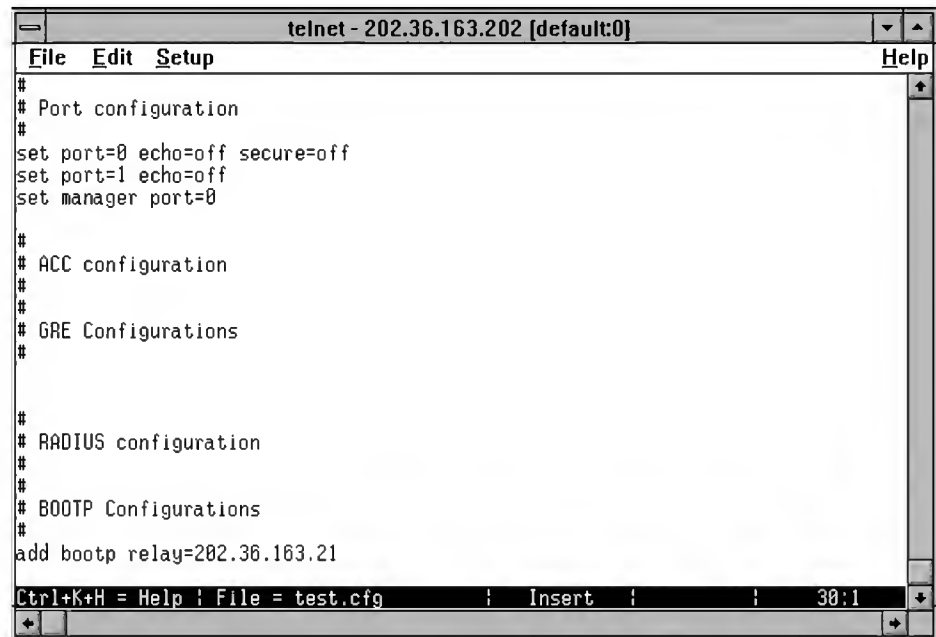
Cursor Movement		Deletion	
↑ or Ctrl/Z	Up one line	Ctrl/T	Delete word right
↓ or Ctrl/X	Down one line	Ctrl/Y	Delete line
→	Right one character		
←	Left one character	Block Operations	
Ctrl/B	Start of file	Ctrl/K,B	Begin block mark
Ctrl/D	End of file	Ctrl/K,D	Unmark block
Ctrl/A	Start of line	Ctrl/K,U	Cut block to buffer
Ctrl/E	End of line	Ctrl/K,C	Copy block to buffer
Ctrl/U	Up one screen	Ctrl/K,V	Paste block from buffer
Ctrl/V	Down one screen	Ctrl/K,Y	Delete block
Ctrl/F	Word right		

Search		Exit	
Ctrl/K,F	Find text	Ctrl/K,X	Exit editor; save file
Ctrl/L	Repeat last find	Ctrl/C	Quit editor; don't save file

Miscellaneous			
Ctrl/I	Insert mode	Ctrl/O	Overstrike mode
Ctrl/W	Refresh the screen	Ctrl/K,H	Display help screen
Ctrl/K,O	Open a file		

The VT100 screen only supports 24 lines, unlike a PC. Lines 1–23 are used to display the text of the file being edited, and line 24 is used as the status bar and command line (Figure 1-7 on page 1-39). The status bar displays the current file name, line and column position in the file and the editing mode (overstrike or insert). When additional command information is required, such as a file name or search text, then a prompt is displayed in the status bar.

Figure 1-7: The editor screen layout.



The editor is invoked with the command:

```
EDIT [filename]
```

The file name is optional as a file can be loaded, or a new file can be created from within the editor itself. The editor is currently limited to editing one file at a time. To overcome this limitation use the cut and paste facility to transfer text between files.



Before starting the editor make sure your terminal, terminal emulation program or Telnet client is 100% compatible with a VT100 terminal.

Help can be obtained at any time while in the editor by pressing [Ctrl/K,H]; that is, holding down the Ctrl key and pressing in turn the K key then the H key.

Examples To edit a file called SHOW SCP, use the command:

```
EDIT SHOW.SCP
```

See Also DELETE FILE
LOAD
SHOW FILE

ENABLE HTTP DEBUG

Syntax `DISABLE HTTP DEBUG={ALL|AUTH|MSG|SESSION}`

Description This command enables HTTP server debugging. Debug output is sent to the terminal session or Telenet connection from which the command was entered.

The DEBUG parameter specifies the type of debugging to be enabled. If AUTH is specified, debugging of authentication attempts is enabled. If MSG is specified, debugging of HTTP GET and SET requests and responses, is enabled. If SESSION is specified, debugging of TCP state changes and session activity is enabled. If ALL is specified, all debugging is enabled. To enable combinations of debugging options, enter multiple commands. Debugging is disabled by default.

Examples To enable debugging of authentication attempts and HTTP GET/SET messages, use the command:

```
ENABLE HTTP DEBUG=AUTH
ENABLE HTTP DEBUG=MSG
```

See Also `DISABLE HTTP DEBUG`
`DISABLE HTTP SERVER`
`ENABLE HTTP SERVER`
`RESET HTTP SERVER`
`SHOW HTTP CLIENT`
`SHOW HTTP DEBUG`
`SHOW HTTP SERVER`
`SHOW HTTP SESSION`

ENABLE HTTP SERVER

Syntax `ENABLE HTTP SERVER`

Description This command enables the HTTP server. The HTTP server serves HTML pages out of the router's FLASH memory to a web browser, and allows users to login into the router and dynamically configure the router via a web-based GUI (*Graphical User Interface*). The server is enabled by default.

Examples To enable the HTTP server, use the command:

```
ENABLE HTTP SERVER
```

See Also `DISABLE HTTP DEBUG`
`DISABLE HTTP SERVER`
`ENABLE HTTP DEBUG`
`RESET HTTP SERVER`
`SHOW HTTP CLIENT`
`SHOW HTTP DEBUG`
`SHOW HTTP SERVER`
`SHOW HTTP SESSION`

ENABLE USER

Syntax `ENABLE USER=login-name`

where:

- *login-name* is a character string, 1 to 32 characters in length. Valid characters are uppercase letters (A–Z), lowercase letters (a–z), and decimal digits (0–9). The string may not contain spaces.

Description This command enables a user login name that has been disabled. The USER parameter specifies the login name for the user. It is case insensitive. Login attempts using the login name will be processed as normal.

See Also ADD USER
 DELETE USER
 DISABLE USER
 PURGE USER
 RESET USER
 SET USER
 SHOW USER

HELP

Syntax `HELP [topic]`

where:

- *topic* is a topic to display.

Description This command displays online help for commands. If a topic is not specified, a list of available topics is displayed. If a topic is specified, and is available, a list of commands relating to the topic is displayed.

The system help file must be assigned using the SET HELP command on page 1-50.

Examples To get help on PPP, use the command:

`HELP PPP`

See Also SET HELP
 SHOW SYSTEM

LOAD

Syntax

```
LOAD [METHOD=TFTP] [DELAY=delay] [DESTINATION=FLASH]
    [FILE=filename] [SERVER=ipadd]

LOAD [METHOD={HTTP|WEB|WWW}] [DELAY=delay]
    [DESTINATION=FLASH] [FILE=filename]
    [PROXYPORT=1..65535] [SERVER=ipadd]

LOAD [METHOD=ZMODEM] [DELAY=delay] [DESTINATION=FLASH]
    [PORT=port]

LOAD [METHOD=NONE] [DELAY=delay] [DESTINATION=FLASH]
    [FILE=filename] [PORT=port]
```

where:

- *delay* is a time delay, in seconds.
- *filename* is the name of the file to load. This may be a full path name for the file in the syntax of the server from which the file will be loaded.
- *ipadd* is an IP address in dotted decimal notation.
- *port* is the number of an asynchronous port. Ports are numbered sequentially starting with port 0.

Description This command downloads a file to the router using *Trivial File Transfer Protocol* (TFTP), *HyperText Transfer Protocol* (HTTP), ZMODEM or direct input from an asynchronous port. Any parameters not specified use the default values set with the SET LOADER command on page 1-51. Some parameters are invalid or have different meanings depending on the method used to download the file.

The DELAY parameter specifies the delay, in seconds, between initiating the file download and the download actually starting. This feature is provided to allow reconfiguration of ports and devices after initiating the download. For example, a manager may be at a remote site with a single PC which is to act as both the access device to the router and the TFTP server. By specifying a delay, the manager has time to reconfigure the PC from terminal emulation mode to TFTP server mode before the download starts. The DELAY parameter is optional.

The DESTINATION parameter specifies where the file will be stored. Only FLASH is valid and the file is stored in the FLASH File System (FFS) on the router. If DESTINATION is not specified, and has not been set with the SET LOADER command on page 1-51, the default is FLASH.

The FILE parameter specifies the name of the file, in the syntax of the server from which the file will be downloaded. The FILE parameter is required unless it has been with the SET LOADER command on page 1-51. The FILE parameter is a full path name rather than just a file name. The only restriction is that the last part of the file parameter must be a valid file name for the LOADER module. When METHOD is set to TFTP, HTTP, ZMODEM or NONE, valid file names are of the form *filename.ext* where *filename* is one to eight characters in length and *ext* is three characters in length. The following are examples of valid file names for methods TFTP, ZMODEM or NONE:

```
\user\public\filename.ext ; UNIX or DOS server
[network.cfg]filename.ext ; DEC VAX server
```

Note that, starting at the end of the file name and working backwards, the first character not valid in file names delimits a valid file name for the router. When METHOD is set to HTTP, the FILE parameter specifies the URL of the file to load. The URL is parsed to extract the remote server address, which may be a fully qualified domain name (e.g. host.company.com) or an IP address and port number (e.g. 192.168.1.1:80). The following are examples of valid file names for method HTTP:

```
http://host.company.com/path/filename.ext
http://192.168.3.4/path/filename.ext
http://192.168.3.4:8000/path/filename.ext
```

The METHOD parameter specifies the method to use when downloading the file. If HTTP is specified, HTTP is used to download the file. The options WEB and WWW are synonyms for HTTP. If TFTP is specified, TFTP is used to download the file. If ZMODEM is specified, the ZMODEM protocol is used to download the file. If ZMODEM is specified, the PORT parameter must be specified, unless it has been set with the SET LOADER command on page 1-51. If NONE is specified, only text files can be downloaded and all input received via the port will be directed to the specified file on the router's file subsystem. The file transfer is terminated by the first control character received that is not a CR or LF character. The FILE parameter is not valid when METHOD is set to ZMODEM. The PORT parameter is not valid when METHOD is set to HTTP, WEB, WWW, TFTP or NONE. The default is TFTP.

The PORT parameter specifies the asynchronous port via which the file will be downloaded, when the METHOD parameter is set to ZMODEM or NONE. If METHOD is set to ZMODEM or NONE, the PORT parameter is required unless it has been set with the SET LOADER command on page 1-51.

The PROXYPORT parameter specifies the port on a proxy server. If METHOD is set to HTTP, WEB or WWW and access to the HTTP server is via a proxy server, the PROXYPORT parameter is required, unless it has been set by the SET LOADER command on page 1-51. The default is 80. The PROXYPORT parameter is not valid if METHOD is set to TFTP, ZMODEM or NONE.

The SERVER parameter specifies the IP address of the TFTP server, HTTP server or proxy server. When METHOD is set to TFTP, the SERVER parameter specifies the IP address of the TFTP server. When METHOD is set to HTTP, and access to the HTTP server is direct (not via a proxy server), if there is no name server set, then the SERVER parameter must specify the IP address of the HTTP server. See SET IP NAMESERVER command on page 6-95 of *Chapter 6, Internet Protocol (IP)* for more information about setting up name servers. When METHOD is set to HTTP and access to the HTTP server is via a proxy server, the SERVER parameter specifies the IP address of the proxy server. The SERVER parameter is required for these methods, unless it has been set by the SET LOADER command on page 1-51. The PING command on page 6-82 of *Chapter 6, Internet Protocol (IP)* can be used to verify that the router can communicate with the server via IP. The SERVER parameter is not valid when METHOD is set to ZMODEM or NONE.

Examples To download a release using the default values set previously with the SET LOADER command on page 1-51, use the command:

```
LOAD
```

To download release 28-761.rel into the FLASH File System from a TFTP server with an IP address of 172.16.8.5, with a delay of one minute, use the command:

```
LOAD FILE=28-761.REL DESTINATION=FLASH SERVER=172.16.8.5
DELAY=60
```

To load a script called SHOW.SCP from asynchronous port 1, use the command:

```
LOAD FILE=SHOW.SCP PORT=1
```

To load the script SHOW.SCP from asynchronous port 1 using the ZMODEM protocol, use the command:

```
LOAD PORT=1 METHOD=ZMODEM
```

To download the file 8-191.rez from the downloads directory on the web server at www.company.com, when a name server has been set, use the command:

```
LOAD METHOD=HTTP DEST=FLASH  
FILE=http:www.company.com/downloads/8-191.rez
```

To download the file 8-191.rez from the downloads directory on the web server at www.company.com (with IP address 192.168.1.1) when a name server is not defined, use the command:

```
LOAD METHOD=HTTP DEST=FLASH  
FILE=http:www.company.com/downloads/8-191.rez  
SERVER=192.168.1.1
```

See Also SET LOADER
SHOW LOADER
UPLOAD

LOGIN

Syntax LOGIN [*login-name*]

where:

- *login-name* is a character string, 1 to 32 characters in length. Valid characters are uppercase letters (A–Z), lowercase letters (a–z), and decimal digits (0–9). The string may not contain spaces.

Description This command is used to login to the router. The User Authentication Facility prompts the user for a login name (if not specified) and a password. The user must enter appropriate responses, pressing [Return] after each response. Characters entered at the password prompt are not echoed to the screen, for security reasons.



The password prompt is displayed regardless of whether or not a password is required for the login name entered by the user. This makes it more difficult for an intruder to discover valid login name/password combinations.

If the user enters an invalid login name or password, the sequence is repeated a set number of times. If a valid login name and password has still not been entered the terminal or Telnet session is *locked out* for a period of time. During this period the password prompt is withheld, preventing the user from logging in or entering commands. The manager can specify the number of login attempts allowed and the length of the lockout period using the SET USER command on page 1-57.

This command is not normally required. The user will automatically be prompted to enter a login name and password when attempting to access the router via Telnet or a terminal connected to an asynchronous port set to SECURE mode.

This command might be used to login from a terminal connected to an asynchronous port that is not in SECURE mode in order to use facilities that are only available to logged in users, or to login as another user in order to acquire different rights, such as MANAGER privilege.

This command may be abbreviated to LOGI. The command LOGON is an alias for LOGIN.



If a user Telnets to the router but does not attempt to login within one minute, the router automatically times out the session and terminates the Telnet connection.

See Also LOGOFF

LOGOFF

Syntax LOGOFF

Description This command is used to log out from the router. For a terminal attached to an asynchronous port, the port returns to its default prompting state, either the login prompt for a port in SECURE mode, or the command prompt. For a Telnet session the TCP connection is terminated. LOGOUT is an alias for the LOGOFF and both commands may be abbreviated to LO.

See Also LOGIN

MODIFY

Syntax MODIFY ADDR=address SIZE={BYTE|LONG|WORD} VAL=value-list
[SPACE={SD|SP|UD|UP|UR}]

where:

- *address* is the base address of the block of memory to modify.
- *value-list* is either a list of up to five numbers (in hexadecimal) separated by commas (e.g. VAL=12,4ac,0,14e,65), or a text string of up to twenty characters surrounded by double quotes (e.g. VAL="string").

Description This command modifies (overwrites) the contents of the router's memory. The values to be written to memory are specified by the VAL parameter and are written to contiguous memory locations starting at the memory address specified by the ADDR parameter. The SIZE parameter specifies whether the values are written as BYTES, LONGWORDS or WORDS. ADDR, VAL and SIZE must be specified. The SPACE parameter is optional and can be used to select any of the valid CPU address spaces (Table 1-5 on page 1-36). If SPACE is not specified the value will default to SD.



It is possible to use this command to modify any memory or I/O devices. This may interrupt the operation of the router.

The MODIFY command is provided mainly as a diagnostic tool. It should not be needed for normal operation of the router.

Examples This example modifies the first two words of memory starting at memory location 0x00000000:

```
MOD ADDR=0 SIZE=WORD VAL=5, 6AA4
```

See Also DUMP

PURGE USER

Syntax PURGE USER

Description This command deletes all users from the User Authentication Database. The MANAGER account remains but the password is set to the default password, "friend". Global configuration parameters and counters, and counters for the MANAGER account, are not affected. To clear these counters use the RESET USER command on page 1-48.

See Also ADD USER
DELETE USER
DISABLE USER
ENABLE USER
RESET USER
SET USER
SHOW USER

RENAME

Syntax RENAME *src-filename dest-filename*

where:

- *src-filename* and *dest-filename* are file identifiers of the form [device:]name.ext. Valid characters are the lowercase letters (a-z), digits (0-9) and the hyphen character (-).

Description This command renames the specified file. The source file name must identify an existing file, and the destination file name must not already be in use. If the source file is not a text file then the source and destination file extensions must be the same.



Caution must be taken when renaming files, such as releases and configurations, since they contain information which is vital to the intended operation of the router.

Examples To rename the file “old.scp” to “new.scp”, use the command:

```
RENAME OLD.SCP NEW.SCP
```

See Also DELETE FILE
SHOW FILE

RESET HTTP SERVER

Syntax RESET HTTP SERVER

Description This command resets the HTTP server. The server is restarted, debugging is disabled and all counters are reset to zero (0).

Examples To reset the HTTP server, use the command:

```
RESET HTTP SERVER
```

See Also DISABLE HTTP DEBUG
DISABLE HTTP SERVER
ENABLE HTTP DEBUG
ENABLE HTTP SERVER
SHOW HTTP CLIENT
SHOW HTTP DEBUG
SHOW HTTP SERVER
SHOW HTTP SESSION

RESET LOADER

Syntax RESET LOADER

Description This command aborts the current file transfer being undertaken by the LOADER module. All resources used by the transfer are released and any file in the process of being created is deleted. The LOADER module becomes immediately ready for a new load to be initiated.

See Also LOAD
SET LOADER
SHOW LOADER

RESET USER

Syntax RESET USER[=*login-name*] [COUNTERS [= {ALL|GLOBAL|USER}]]

where:

- *login-name* is a character string, 1 to 32 characters in length. Valid characters are uppercase letters (A–Z), lowercase letters (a–z), and decimal digits (0–9). The string may not contain spaces.

Description This command is used to reset User Authentication Database counters for one or all users, or to reset global counters for the User Authentication Facility.

If a login name is specified with the USER parameter, the COUNTERS parameter is optional (only USER may be specified) and the activity counters for the specified user are reset. The login name is not case sensitive.

If a login name is not specified with the USER parameter then the COUNTERS parameter is required and specifies which counters should be reset. If USER is specified, the activity counters for all users are reset. If GLOBAL is specified, the global counters for the User Authentication Facility are reset. If ALL is specified, all counters are reset.

Examples To reset the activity counters for user BRUCE, use the command:

```
RESET USER=BRUCE
```

To reset the activity counters for all users, use the command:

```
RESET USER COUNTERS=USER
```

To reset the global counters, use the command:

```
RESET USER COUNTERS=GLOBAL
```

See Also ADD USER
DELETE USER
DISABLE USER
ENABLE USER
PURGE USER
SET USER
SHOW USER

RESTART

Syntax RESTART {REBOOT|ROUTER} [CONFIG=*filename*|NONE]

where:

- *filename* is a file name of the form `device:filename.ext`. Valid characters are the lowercase letters (a–z), digits (0–9) and the hyphen character (-). Wildcards are not allowed.

Description This command restarts the router with either the current configuration file (set with the SET CONFIG command on page 1-49) or the specified configuration file.

If REBOOT is specified the router performs a cold start (hardware reset) and executes the default configuration file, if one is defined. The CONFIG parameter may not be specified.

If ROUTER is specified the router performs a warm start of all software modules (the hardware is not reset) and executes the default configuration file, if one is defined. The CONFIG parameter may be used to specify a script or configuration file other than the current default. The file extension must be “`scp`” or “`cfg`”. If NONE is specified, the router will reboot without executing any configuration file.

Examples To restart the router using the configuration file `test.cfg` instead of the default configuration file, use the command:

```
RESTART ROUTER CONFIG=TEST.CFG
```

See Also SHOW CONFIG
SHOW EXCEPTION
SHOW STARTUP

SET CONFIG

Syntax SET CONFIG=*filename*

where:

- *filename* is a file name of the form `device:filename.ext`. Valid characters are the lowercase letters (a–z), digits (0–9) and the hyphen character (-). Wildcards are not allowed.

Description This command sets the script file which the router will use as its default configuration. The file name is stored in a FLASH File System file.

The CONFIG parameter specifies the name of the script or configuration file to use. The file extension must be “`scp`” or “`cfg`”. The file must already exist on the router. The commands in the script file are executed when the router is rebooted or performs a warm restarted.

Examples To set the default configuration file to boot.cfg, use the command:

```
SET CONFIG=BOOT.CFG
```

See Also RESTART
CREATE CONFIG
SHOW CONFIG

SET HELP

Syntax SET HELP=*helpfile*

where:

- *helpfile* is a file name of the form `device:filename.ext`. Valid characters are the lowercase letters (a–z), digits (0–9) and the hyphen character (-). Wildcards are not allowed.

Description This command sets the system help file used by the HELP command on page 1-41. The HELP parameter specifies the name of the text file containing the help text for the router. If the device field is not specified, the default is FLASH.

Examples To set the help file to the file E72-01.HLP, use the command:

```
SET HELP=E72-01.HLP
```

See Also HELP
SHOW SYSTEM

SET INSTALL

Syntax SET INSTALL={TEMPORARY|PREFERRED|DEFAULT}
[RELEASE={*release-name*|EPROM}]

where:

- *release-name* is the name of a release file, of the form `device:filename.ext`. Valid characters are the lowercase letters (a–z), digits (0–9) and the hyphen character (-). Wildcards are not allowed.

Description This command sets up release information for one of the installs.

The INSTALL parameter specifies which install is to be set. The INSTALL module is responsible for maintaining install information and loading the correct install at boot. An *install* is a record identifying a release. Three installs are maintained by the INSTALL module, *temporary*, *preferred* and *default*.

The default install is the install of last resort. The release for the default install can not be changed by the manager and is always the EPROM release.

The temporary and preferred installs are completely configurable. The release may be EPROM or a release stored in FFS.

The RELEASE parameter specifies the release file for this install. The release file is either a file name of the form `device:filename.ext` for files in the file subsystem, or EPROM, to indicate the EPROM release. The default value for the device field is FLASH.

Examples To set up the release file 28-761.rel in FLASH as a temporary install, use the command:

```
SET INSTALL=TEMPORARY RELEASE=28-761.REL
```

See Also DELETE INSTALL
SHOW INSTALL

SET LOADER

Syntax SET LOADER [DELAY=*delay*] [DESTINATION=FLASH]
[FILE=*filename*] [METHOD={HTTP|TFTP|WEB|WWW|ZMODEM|NONE}]
[PORT=*port*] [PROXYPORT=*proxyport*] [SERVER=*ipadd*]

where:

- *filename* is the default name of the file to load. This may be a full path name for the file in the syntax of the server from which the file will be loaded.
- *ipadd* is an IP address in dotted decimal notation.
- *delay* is a time delay, in seconds.

Description This command sets default values for the LOAD command on page 1-42. All values that can be specified with the LOAD command can also be specified as defaults with the SET LOADER command. Any parameters not specified on the LOAD command will use the default value set by the SET LOADER command.

The DELAY parameter specifies the delay, in seconds, between initiating the file download and the download actually starting. This feature is provided to allow reconfiguration of ports and devices after initiating the download. For example, a manager may be at a remote site with a single PC which is to act as both the access device to the router and the TFTP server. By specifying a delay, the manager has time to reconfigure the PC from terminal emulation mode to TFTP server mode before the download starts. The DELAY parameter is optional.

The DESTINATION parameter specifies where the file will be stored. Only FLASH is valid and the file is stored in the FLASH File System (FFS) on the router. If DESTINATION is not specified, and has not been set with the SET LOADER command on page 1-51, the default is FLASH.

The FILE parameter specifies the name of the file, in the syntax of the server from which the file will be downloaded. The FILE parameter is required unless it has been with the SET LOADER command on page 1-51. The FILE parameter is a full path name rather than just a file name. The only restriction is that the last part of the file parameter must be a valid file name for the LOADER

module. When METHOD is set to TFTP, HTTP, ZMODEM or NONE, valid file names are of the form `filename.ext` where `filename` is one to eight characters in length and `ext` is three characters in length. The following are examples of valid file names for methods TFTP, ZMODEM or NONE:

```
\user\public\filename.ext ; UNIX or DOS server  
[network.cfg]filename.ext ; DEC VAX server
```

Note that, starting at the end of the file name and working backwards, the first character not valid in file names delimits a valid file name for the router. When METHOD is set to HTTP, the FILE parameter specifies the URL of the file to load. The URL is parsed to extract the remote server address, which may be a fully qualified domain name (e.g. `host.company.com`) or an IP address and port number (e.g. `192.168.1.1:80`). The following are examples of valid file names for method HTTP:

```
http://host.company.com/path/filename.ext  
http://192.168.3.4/path/filename.ext  
http://192.168.3.4:8000/path/filename.ext
```

The METHOD parameter specifies the method to use when downloading the file. If HTTP is specified, HTTP is used to download the file. The options WEB and WWW are synonyms for HTTP. If TFTP is specified, TFTP is used to download the file. If ZMODEM is specified, the ZMODEM protocol is used to download the file. If ZMODEM is specified, the PORT parameter must be specified, unless it has been set with the SET LOADER command on page 1-51. If NONE is specified, only text files can be downloaded and all input received via the port will be directed to the specified file on the router's file subsystem. The file transfer is terminated by the first control character received that is not a CR or LF character. The FILE parameter is not valid when METHOD is set to ZMODEM. The PORT parameter is not valid when METHOD is set to HTTP, WEB, WWW, TFTP or NONE. The default is TFTP.

The PORT parameter specifies the asynchronous port via which the file will be downloaded, when the METHOD parameter is set to ZMODEM or NONE. If METHOD is set to ZMODEM or NONE, the PORT parameter is required unless it has been set with the SET LOADER command on page 1-51.

The PROXYPORT parameter specifies the port on a proxy server. If METHOD is set to HTTP, WEB or WWW and access to the HTTP server is via a proxy server, the PROXYPORT parameter is required, unless it has been set by the SET LOADER command on page 1-51. The default is 80. The PROXYPORT parameter is not valid if METHOD is set to TFTP, ZMODEM or NONE.

The SERVER parameter specifies the IP address of the TFTP server, HTTP server or proxy server. When METHOD is set to TFTP, the SERVER parameter specifies the IP address of the TFTP server. When METHOD is set to HTTP, and access to the HTTP server is direct (not via a proxy server), if there is no name server set, then the SERVER parameter must specify the IP address of the HTTP server. See SET IP NAMESERVER command on page 6-95 of *Chapter 6, Internet Protocol (IP)* for more information about setting up name servers. When METHOD is set to HTTP and access to the HTTP server is via a proxy server, the SERVER parameter specifies the IP address of the proxy server. The SERVER parameter is required for these methods, unless it has been set by the SET LOADER command on page 1-51. The PING command on page 6-82 of *Chapter 6, Internet Protocol (IP)* can be used to verify that the router can communicate with the server via IP. The SERVER parameter is not valid when METHOD is set to ZMODEM or NONE.

Examples To set the default download parameters to be release 28-72.rel downloaded into the FLASH File System from the TFTP server with IP address 172.16.8.5, with a delay of one minute, use the command:

```
SET LOAD FILE=28-72.REL DESTINATION=FLASH SERVER=172.16.8.5
DELAY=60
```

See Also LOAD
SHOW LOADER

SET MANAGER PORT

Syntax SET MANAGER PORT={*port-number*|NONE}

where:

- *port-number* is the number of the port. Ports are numbered sequentially starting with port 0.

Description This command sets the semipermanent manager port. If a valid port number is specified the port becomes the semipermanent manager port. If the specified port was secure before the command was entered it loses its secure setting. If any other port is currently the semipermanent manager port then that port loses its semipermanent MANAGER privilege and becomes a secure port. If NONE is specified the current semipermanent manager port (if any) loses its semipermanent MANAGER privilege and becomes a secure port. There may be no more than one semipermanent manager port at any time.



This command is one of the security commands (see “Database Security” on page 1-9). If the security timer expires before the command is entered, the manager will be prompted to re-enter the password for the login name from which the command was issued.

Examples To set port 0 as the semipermanent manager port, use the command:

```
SET MANAGER PORT=0
```

To remove the semipermanent manager port, use the command:

```
SET MANAGER PORT=NONE
```

See Also LOGIN
SHOW MANAGER PORT
SET PORT in *Chapter 2, Interfaces*

SET PASSWORD

Syntax SET PASSWORD

Description This command changes the login password for the user currently logged in to the port from which the command was entered. If a user is not logged in to the port an error message is displayed. If a user is logged in to the port, the user is prompted for the existing password, the new password and confirmation of the new password. The passwords entered are not echoed to the screen.

The new password and the confirmation must be identical for the change to take affect. This reduces the chances of a typing error causing the password to be different from what the user intended.

A log message is generated whenever the password for an account with MANAGER privilege is changed. A configurable minimum password length is enforced. The default is 6 characters.

Examples To change the password for the current user, use the command:

```
SET PASSWORD
Old password:
New password:
Confirm:
```

See Also ADD USER
SET USER

SET SYSTEM CONTACT

Syntax SET SYSTEM CONTACT=*contact-name*

where:

- *contact-name* is a character string, 1 to 256 characters in length. Valid characters are any printable character. If the string includes spaces it must be enclosed in double quotes.

Description This command assigns a string defining the contact name for this router. For example "Bruce Johns, 64-3-343-0803". The string can be a maximum of 80 characters. The text is displayed in the output of the SHOW SYSTEM command on page 1-78. It also updates the MIB object *sysContact* which can then be read using SNMP.

Examples To set the contact name for this router to "Bruce Johns, 64-3-343-0803", use the command:

```
SET SYSTEM CONTACT="Bruce Johns, 64-3-343-0803"
```

See Also SET SYSTEM LOCATION
SET SYSTEM NAME
SET SYSTEM TERRITORY
SHOW SYSTEM

SET SYSTEM LOCATION

Syntax SET SYSTEM LOCATION=*location*

where:

- *location* is a character string, 1 to 256 characters in length. Valid characters are any printable character. If the string includes spaces it must be enclosed in double quotes.

Description This command assigns a string defining the physical location of this router. For example "Laboratory, First Floor, Head Office Building". The string can be a maximum of 80 characters. The text is displayed in the output of the SHOW SYSTEM command on page 1-78. It also updates the MIB object *sysLocation* which can then be read using SNMP.

Examples To set the location for this router to "Laboratory, First Floor, Head Office Building", use the command:

```
SET SYSTEM LOCATION="Laboratory, First Floor, Head Office
Building"
```

See Also SET SYSTEM CONTACT
SET SYSTEM NAME
SET SYSTEM TERRITORY
SHOW SYSTEM

SET SYSTEM NAME

Syntax SET SYSTEM NAME=*name*

where:

- *name* is a character string, 1 to 256 characters in length. Valid characters are any printable character. If the string includes spaces it must be enclosed in double quotes.

Description This command assigns a string defining the name of this router. By convention this is the full domain name of the IP entity. For example, *nd1.co.nz*. The name can be a maximum of 80 characters. The text is displayed in the output of the SHOW SYSTEM command on page 1-78. It also updates the MIB object *sysName* which can then be read using SNMP.

Examples To set the name for this router to "nd1.co.nz", use the command:

```
SET SYSTEM NAME="nd1.co.nz"
```

See Also SET SYSTEM CONTACT
SET SYSTEM LOCATION
SET SYSTEM TERRITORY
SHOW SYSTEM

SET SYSTEM TERRITORY

Syntax SET SYSTEM TERRITORY={AUSTRALIA|CHINA|EUROPE|JAPAN|KOREA|NEWZEALAND|USA}

Description This command assigns a territory identifier for the router. The territory identifier is used by the Q.931 and PBX modules to set defaults that are appropriate for the territory in which the router is being operated. The default territory is EUROPE.



If the router territory identifier is changed, parameters in the Q.931 and PBX modules that are influenced by the territory in which the router is being operated will automatically be changed to values appropriate for the new territory setting. If the current territory value is specified, i.e. the territory is unchanged, then the module parameters are restored to the default values for that territory.

Examples To set the name for this router to Australia, use the command:

```
SET SYSTEM TERRITORY=AUSTRALIA
```

See Also SET SYSTEM CONTACT
SET PBX in *Chapter 14, Telephony Services*
SET Q931 in *Chapter 4, Integrated Services Digital Network (ISDN)*
SET SYSTEM LOCATION
SET SYSTEM NAME
SHOW PBX in *Chapter 14, Telephony Services*
SHOW Q931 in *Chapter 4, Integrated Services Digital Network (ISDN)*
SHOW SYSTEM

SET TIME

Syntax SET [TIME=*time*] [DATE=*date*]

where:

- *time* is the time in 24 hour format (hh:mm:ss).
- *date* is the date in the format dd-mm-yy where the month is given as the first three letters of the month name (e.g. APR).

Description This command sets the time and/or date stored in the router's real-time clock.

Examples The following commands set the router's real-time clock to 10pm on 29 January 1993:

```
SET TIME=22:00:00
SET DATE=29-JAN-93
```

See Also SHOW TIME

SET USER

Syntax `SET USER=login-name [CBNUMBER=e164number]
 [DESCRIPTION=description] [PASSWORD=password]
 [PRIVILEGE={USER|MANAGER}] [TELNET={YES|NO}]`

`SET USER [LOGINFAIL=1..10] [LOCKOUTPD=0..30000]
 [MANPWDFAIL=1..5] [SECUREDELAY=10..600]
 [MINPWDLEN=1..23]`

where:

- *login-name* is a character string, 1 to 32 characters in length. Valid characters are uppercase letters (A–Z), lowercase letters (a–z), and decimal digits (0–9). The string may not contain spaces.
- *password* is a character string, 1 to 32 characters in length. Valid characters are any printable character. If the string contains spaces it must be enclosed in double quotes.
- *e164number* is the phone number to dial when performing callback. It may contain digits (0–9) and should be a valid phone number as described in CCITT standard E.164.
- *description* is a character string, 1 to 23 characters in length. Valid characters are any printable character. If the string contains spaces it must be enclosed in double quotes.

Description This command modifies a user record in the User Authentication Database or alters global parameters affecting the User Authentication Facility.

The first variant of the command is used to alter a user record in the User Authentication Database. The USER parameter specifies the login name of a user in the database. Other parameters specified on the command modify the information stored in the database for that user. The second variant of the command is used to alter the global security parameters for the User Authentication Facility.

The PASSWORD parameter specifies the password for the user. The password is case sensitive. It is intended that the PASSWORD parameter be used to set an initial password for the user and that the user will change it to some string known only to the user, using the command:

```
SET PASSWORD
```

A password set with the SET PASSWORD command on page 1-54 may contain any printing character. A configurable minimum password length is enforced. The default is 6 characters.

The CBNUMBER parameter specifies the ISDN phone number to use when making a call back to a remote user using the PPP callback facility.

The DESCRIPTION parameter specifies a descriptive text for the entry, such as the full name and location of the user. This string may contain any printing character and the case is preserved in output.

The PRIVILEGE parameter specifies the privilege level for the user. The default is USER. A user with USER privilege has access to only a limited subset of commands, generally commands that only affect the user's own session or

asynchronous port. A user with MANAGER privilege has access to the complete router command set.

The TELNET parameter specifies whether or not the user is permitted to use the TELNET command on page 7-11 of *Chapter 7, Terminal Server* to Telnet to another host when logged in via Telnet.

The LOGINFAIL parameter sets the number of successive login failures a user may make before the login prompt is withheld for the lockout period. The default value is 3.

The LOCKOUTPD parameter sets the number of seconds that the login prompt will be withheld when the number of login retries exceeds the value set by LOGINFAIL. The default is 600 seconds.

The MANPWDFAIL parameter sets the number of successive attempts a manager may make to enter the correct password while entering a security command before the session is automatically logged off. The default value is 3.

The SECUREDELAY parameter sets the number of seconds that may elapse between the entry of one security command and the next without the user being required to re-enter the password to validate the command. The default is 60 seconds.

The MINPWDLEN parameter sets the minimum password length that will be enforced for the ADD USER commands and SET PASSWORD commands. The default is 6 characters.

Examples To change the password to "BZ4gal" and the privilege level to MANAGER for user BRUCE, use the command:

```
SET USER=BRUCE PASSWORD=BZ4gal PRIVILEGE=MANAGER
```

To change the minimum password length to eight characters for all users, use the command:

```
SET USER MINPWDLEN=8
```

See Also ADD USER
DELETE USER
DISABLE USER
ENABLE USER
PURGE USER
RESET USER
SHOW USER

SHOW ALIAS

Syntax SHOW ALIAS

Description This command displays the aliases currently defined on the router (Table 1-8 on page 1-59, Table 1-7 on page 1-59).

Figure 1-8: Example output from the SHOW ALIAS command.

```
Alias ..... df
String .... delete file=1-190.rez

Alias ..... ii
String .... ip interface
```

Table 1-7: Parameters displayed in the output of the SHOW ALIAS command.

Parameter	Meaning
Alias	The name of the alias.
String	The string substituted for the alias when it appears in a command line.

See Also ADD ALIAS
 DELETE ALIAS

SHOW BUFFER

Syntax SHOW BUFFER [SCAN[=*address* [QUEUEPOINTERS]]]

where:

- *address* is the memory address of a section of router code, expressed in hexadecimal.

Description This command displays information about the memory buffers is use by router modules. If no optional parameters are specified, a summary of the buffers in use is displayed (Figure 1-9 on page 1-60, Table 1-8 on page 1-60).

The SCAN parameter displays more detailed information about buffers usage. If an address is not specified, the memory addresses of sections of router code and the number of buffers in used by that section are displayed (Figure 1-10 on page 1-60). If an address is specified, the addresses of the buffers in use by that section of router code are displayed (Figure 1-11 on page 1-61). The value for *address* is obtained from the output of a previous SHOW BUFFER SCAN command.

The QUEUEPOINTERS parameter displays additional information about the contents of the buffers used by the router code section at the specified address (Figure 1-12 on page 1-61), and is only valid when the SCAN parameter is specified with a valid address.



The SCAN and QUEUEPOINTERS parameters display low-level debugging information. Use these parameters only when directed to by technical support personnel.

Figure 1-9: Example output from the SHOW BUFFER command.

```

Memory ( DRAM ) ..... 4096 kB
Free Memory ..... 40 %
Free buffers ..... 947
Total buffers ..... 1059
Buffer level 3 ..... 211 (don't process input frames)
Buffer level 2 ..... 423 (don't do monitor or command output)
Buffer level 1 ..... 741 (don't buffer up log messages)

```

Table 1-8: Parameters displayed in the output of the SHOW BUFFER command.

Parameter	Meaning
Memory (DRAM)	The total amount of DRAM installed in the router.
Free memory	The amount of free (unused) memory, as a percentage of total available memory.
Free buffers	The number of free (unused) memory buffers.
Total buffers	The total number of memory buffers.
Buffer level n	Levels at which certain processes are halted if the value of "Free buffers" drops below that level.

Figure 1-10: Example output from the SHOW BUFFER SCAN command.

```

Scan of buffers in use

00093d62    2  001338a2    1  0013d27c    1  000cd26a    1  000ccfc2    7
000cd326    5  000cd542    1  0006d1f0    1  000a03e4    1  000a4256    1
001f544e    1  001f5484    1  001f54c0    1  000a50da    1  00082e52    1
0013fe40    2  0008c8b0    1  0008c8f0    1  0008c92c    1  0008f7f6    1
000ebd32    1  000ec0a2    2  000ec364    3  00080048    8  00081352    1
0016ef96    1  0012fd76    1  0012f64a    1  00086e3c    1  0008871a    1
000b6866    1  001f5338   10  001526e0    1  0011e892    2  00099486    1
001194d4    1  0011deb0   17  0011fd6a    2  0011d278    1  001139a4    1
0011b354    1  0011d7e8    1  001fe0ca    1  001fb446    1  001fb48c    2
001fb4e8    2  001fb52a    1  0005e95c    1  0005e9f8    1  000d3976    1
00161596    1  00153b60    1  000994ae    1  000d133e    1  000bbc3a    1
00163154    1  001069fc    1  000a4916    1  000a5298    1  00141e26    1
00157156    1  000f4028    1  00169bd8    1  000a9654    1  001352a4   16
000892ae    1  001524fa    1  00087014    1  00089666    1  0008625c    1
0012f6d2    1  00141e30    1  00141e3a    1  0014190e    1  00141940    1
000c512a   15  00087624    1

Total buffers in use - 333

Memory ( DRAM ) ..... 8192 kB
Free Memory ..... 61 %
Free buffers ..... 2860
Total buffers ..... 3193
Buffer level 3 ..... 638 (don't process input frames)
Buffer level 2 ..... 1277 (don't do monitor or command output)
Buffer level 1 ..... 2235 (don't buffer up log messages)

```

Figure 1-11: Example output from the SHOW BUFFER SCAN command for a specified address.

```

002c93bc 002ce7bc 002d42bc 002d49bc 002d57bc 002d5ebc
002d65bc 002df8bc 002dffbc 002e0dbc 002e14bc 002eaebc
002eb5bc 002ec3bc 002ecabc

Memory ( DRAM ) ..... 8192 kB
Free Memory ..... 61 %
Free buffers ..... 2860
Total buffers ..... 3193
Buffer level 3 ..... 638 (don't process input frames)
Buffer level 2 ..... 1277 (don't do monitor or command output)
Buffer level 1 ..... 2235 (don't buffer up log messages)

```

Figure 1-12: Example output from the SHOW BUFFER SCAN QUEUEPOINTERS command.

```

002c93bc 002df8bc 002d5ebc 002c9434 002ce7bc 002e0dbc 002dffbc 002ce834
002d42bc 002d49bc 002569f0 002d4334 002d49bc 002d57bc 002d42bc 002d4a34
002d57bc 002d5ebc 002d49bc 002d5834 002d5ebc 002c93bc 002d57bc 002d5f34
002d65bc 002ec3bc 002eb5bc 002d6634 002df8bc 002dffbc 002c93bc 002df934
002dffbc 002ce7bc 002df8bc 002e0034 002e0dbc 002e14bc 002ce7bc 002e0e34
002e14bc 002eaebc 002e0dbc 002e1534 002eaebc 002eb5bc 002e14bc 002eaf34
002eb5bc 002d65bc 002eaebc 002eb634 002ec3bc 002ecabc 002d65bc 002ec434
002ecabc 002569f0 002ec3bc 002ecb34

Memory ( DRAM ) ..... 8192 kB
Free Memory ..... 61 %
Free buffers ..... 2860
Total buffers ..... 3193
Buffer level 3 ..... 638 (don't process input frames)
Buffer level 2 ..... 1277 (don't do monitor or command output)
Buffer level 1 ..... 2235 (don't buffer up log messages)

```

SHOW CONFIG

Syntax `SHOW CONFIG [DYNAMIC [=module-id]]`

where:

- *module-id* is the name of a router module (see “Module Identifiers and Names” on page B-2 of *Appendix B, Reference Tables* for a complete list).

Description This command displays the current configuration file for the router, or the current dynamic configuration for the router or specified software module.

If no optional parameters are specified, the current default configuration file (set with the SET CONFIG command on page 1-49) is displayed, along with information about how the current configuration in the router was obtained (Figure 1-13 on page 1-62, Table 1-9 on page 1-62).

The DYNAMIC parameter displays the current dynamic configuration of the router, or of the specified software module. The information displayed is the

sequence of router commands required to recreate the current dynamic configuration.

Figure 1-13: Example output from the SHOW CONFIG command.

```
Boot configuration file: boot.cfg (exists)
Current configuration: boot.cfg
```

Table 1-9: Parameters displayed in the output of the SHOW CONFIG command.

Parameter	Meaning
Boot configuration file	<p>The current boot configuration file set with the SET CONFIG command on page 1-49, and whether or not the file exists; one of:</p> <p>"Not set": The boot configuration file has not been set</p> <p>"<filename> (exists)": The boot configuration file has been set to <filename> and <filename> exists.</p> <p>"<filename> (doesn't exist)": The boot configuration file has been set to <filename> but <filename> does not exist.</p>
Current configuration	<p>The source of the current configuration; one of:</p> <p>"None": The router booted up with no configuration, because there was no configuration file set, the file <code>boot.cfg</code> was not found, or the user entered "s" or "S" in response to the "Force EPROM download" message.</p> <p>"<filename> (warm restart)": The router booted up using <filename>, but this was a warm restart (RESTART ROUTER CONF=<filename>).</p> <p>"None (file not found)": The router booted up with no configuration because the required configuration file was not found. Note that RESTART ROUTER CONF=<filename> and SET CONF=<filename> check that the file exists, but it is possible to execute a SET CONF command, and then delete the file!</p> <p>"<filename>": The router booted from the <filename> configuration file. This is the normal case.</p> <p>"<file> (default)": The router booted from the default configuration file, <code>boot.cfg</code>, because a configuration file has not been set. The router looks for <code>boot.cfg</code> in FLASH.</p>

Examples To display the default configuration file, use the command:

```
SHOW CONFIG
```

To display the current dynamic configuration of the router, use the command:

```
SHOW CONFIG DYNAMIC
```

See Also RESTART
CREATE CONFIG
SET CONFIG

SHOW CPU

Syntax `SHOW CPU`

Description This command displays CPU utilisation since the router last restarted (Figure 1-14 on page 1-63, Table 1-10 on page 1-63).

Figure 1-14: Example output from the SHOW CPU command.

```
CPU Utilisation ( as a percentage )
-----
Maximum since router restarted ..... 62
Average since router restarted ..... 0
Average over last minute ..... 0
Average over last 10 seconds ..... 2
Average over last second ..... 1
-----
```

Table 1-10: Parameters displayed in the output of the SHOW CPU command.

Parameter	Meaning
Maximum since router restarted	The maximum CPU utilisation recorded since the router restarted.
Average since router restarted	The average CPU utilisation recorded since the router restarted, as a percentage of total CPU capacity.
Average over last minute	The average CPU utilisation over the last minute, as a percentage of total CPU capacity.
Average over last 10 seconds	The average CPU utilisation over the last 10 seconds, as a percentage of total CPU capacity.
Average over last second	The average CPU utilisation over the last second, as a percentage of total CPU capacity.

See Also `SHOW BUFFER`

SHOW DEBUG

Syntax `SHOW DEBUG`

Description This command displays a snapshot of the state of the router immediately prior to the last fatal condition, and is used for debugging purposes. It generates the same output as the following sequence of commands, in addition to a stack dump:

```
SHOW SYSTEM
SHOW FILES
SHOW INSTALL
SHOW CONFIGURATION DYNAMIC
SHOW BUFFER SCAN
SHOW CPU
SHOW LOG
```

```
SHOW EXCEPTION
SHOW FFILE CHECK
```

See Also SHOW EXCEPTION
SHOW LOG in *Chapter 12, Logging Facility*
SHOW STARTUP
SHOW SYSTEM

SHOW EXCEPTION

Syntax SHOW EXCEPTION

Description This command displays the router exception list (Figure 1-15 on page 1-65).

There may be up to ten entries in the list, ordered from most recent (event 01) to least recent (event 10). The explicit format of each entry depends on the exception type and hence what information was stored for that event.

The *Spurious interrupts* field is the number of spurious interrupts handled by the router since startup. Under normal operating conditions this field should always be zero (0).

The fatal trap with error code of \$001e shown as exceptions 1–4 and 7 is a CPU software trap that is invoked in response to the RESTART command on page 1-49 and hence should not be considered an error.

Figure 1-15: Example output from the SHOW EXCEPTION command.

```

Spurious interrupts = 0

Router exception list
-----
No: 01
  Offset/Type : $008/Bus error      Address   : $0019aaee
  Time        : 09:17:19 on 10-May-1997  Clock Log : 09:16:42 on 10-May-1997
  SSW         : $0225                Fault Addr : $0d0a0044

No: 02
  Offset/Type : $008/Bus error      Address   : $0019aaee
  Time        : 09:15:26 on 10-May-1997  Clock Log : 09:14:29 on 10-May-1997
  SSW         : $0225                Fault Addr : $0d0a0044

No: 03
  Offset/Type : $028/Line A emulator Address   : $0009624c
  Time        : 10:42:59 on 01-May-1997  Clock Log : 10:41:22 on 01-May-1997

No: 04
  Offset/Type : $028/Line A emulator Address   : $0009624c
  Time        : 10:42:59 on 01-May-1997  Clock Log : 10:41:22 on 01-May-1997

No: 05
  Offset/Type : $028/Line A emulator Address   : $0009624c
  Time        : 10:42:59 on 01-May-1997  Clock Log : 10:41:22 on 01-May-1997

No: 06
  Offset/Type : $028/Line A emulator Address   : $0009624c
  Time        : 10:42:59 on 01-May-1997  Clock Log : 10:41:22 on 01-May-1997
-----

```

SHOW FFILE

Syntax SHOW FFILE[=*file-identifier*] [CHECK]

where:

- *file-identifier* is a valid FFS file identifier of the form `device:filename.ext`. Valid characters are the lowercase letters (a-z), digits (0-9) and the hyphen character (-). Wildcards are allowed in any of the elements.

Description This command displays a list of the files in the FFS that match the specified file identifier (Figure 1-16 on page 1-66, Table 1-11 on page 1-66). If a file identifier is not specified then all files are displayed. Wildcards can be used to replace any part of the file identifier to allow a more selective display. The CHECK parameter specifies that the file data checksums are to be verified.



If the CHECK parameter is specified then the command output may take a number of seconds to complete when larger files are being checked.

Figure 1-16: Example output from the SHOW FFILE command.

dev	creator	name	type	size	file date & time	address	check
flash		aa	cfg	1040	06-May-1997 10:55:31	01E09AA8	-
flash		test	cfg	899	03-Jun-1997 15:38:34	01CC8C6C	-
flash		test1	cfg	1768	01-Jun-1997 00:23:52	01E090F4	-
flash		test3	cfg	2501	08-May-1997 11:44:04	01E0AD50	-
flash		b8	scp	3606	06-May-1997 16:43:59	01E09EF8	-
flash		isdn-d	scp	189	01-Jun-1997 00:27:49	01E0981C	-
flash		mtimea	scp	203	28-Apr-1997 15:09:32	01E0991C	-
flash	inst	release	lic	64	05-May-1997 17:30:45	01E09A28	-
flash	load	28-74ang	pat	36960	01-Jun-1997 00:08:32	01E00054	-
flash	load	28-74tst	pat	10676	23-May-1997 17:18:31	01CC4274	-
flash	load	28-74ang	rel	2019228	13-May-1997 15:50:52	01E0BDE0	-
flash	load	28-74	rez	832632	14-May-1997 20:47:05	01FF8DBC	-

flash use:							
	files	2910628 bytes	(12 files)			
	garbage	...	9868 bytes				
	free	1273808 bytes				
	total	4194304 bytes				

Table 1-11: Parameters displayed in the output of the SHOW FFILE command.

Parameter	Meaning
dev	The device in which the file is stored.
creator	The module which created the file.
name	The file name.
type	The file type.
size	The size of the file in bytes, as a decimal number.
file date & time	The date and time the file was created.
address	The base address of the file, in hexadecimal.
check	The result of the file data check (if CHECK was specified).
files	The number of bytes of FLASH memory used by valid files.
garbage	The number of bytes of FLASH memory used by deleted files.
free	The number of bytes of FLASH memory free.
total	The total size of FLASH memory.

Examples To display all the release files created by the Loader module, use the command:

```
SHOW FFILE=FLASH:*.REZ
```

See Also CREATE FFILE
DELETE FFILE

SHOW FILE

Syntax `SHOW FILE [=filename]`

where:

- *filename* is a file identifier of the form `[device:]name.ext`. Valid characters are the lowercase letters (a–z), digits (0–9) and the hyphen character (-). Wildcards are allowed in the name and extension elements.

Description This command displays a list of the files in the file subsystem that match the specified file name (Figure 1-17 on page 1-67, Table 1-12 on page 1-67). Wildcards can be used to replace any part of the file identifier to allow a more selective display. If the file name matches an explicit file and the file is an ASCII text file, the contents of the file are displayed.

Figure 1-17: Example output from the SHOW FILE command.

Filename	Device	Size	Created
28-72.pat	flash	111764	05-May-1997 12:41:42
28-74ang.rel	flash	2013756	09-May-1997 15:58:55
28f72-06.pat	flash	123268	18-Apr-1997 15:58:16
release.lic	flash	32	08-May-1997 16:43:49
test.cfg	flash	1698	09-May-1997 10:39:42

Table 1-12: Parameters displayed in the output of the SHOW FILE command.

Parameter	Meaning
Filename	The name of the file.
Device	The device on which the file is physically stored; "flash".
Size	The size of the file in bytes, as a decimal number.
Created	The date and time the file was created.

Examples To display the contents of the script file CONFIG.SCP, use the command:

```
SHOW FILE=CONFIG.SCP
```

See Also DELETE FILE

SHOW FLASH

Syntax `SHOW FLASH [FFS]`

Description This command displays general status information about the FLASH File System (FFS). The FFS provides a consistent file-based interface to the physical FLASH memory structure, and housekeeping and management functions (Figure 1-18 on page 1-68, Table 1-13 on page 1-68).

Figure 1-18: Example output from the SHOW FLASH command.

```

FFS info:
global operation ..... none
compaction count ..... 35
est compaction time ... 48 seconds
files ..... 328 bytes (3 files)
garbage ..... 655424 bytes
free ..... 1441400 bytes
total ..... 2097152 bytes

diagnostic counters:
event      successes      failures
-----
get         0              0
open        0              1
read        0              0
close       0              0
complete    0              0
write       0              0
create      0              0
put         0              0
delete      0              0
check       0              0
erase       0              0
compact     0              0
verify      0              0
-----

```

Table 1-13: Parameters displayed in the output of the SHOW FLASH command.

Parameter	Meaning
global operation	The global operation currently running; one of "none", "restarting", "erasing", "compacting", or "verifying".
compaction count	The number of times the FLASH has been compacted since the last total erasure.
est compaction time	Estimate of how long compaction would take if it was started now.
files	Amount of space used by valid files.
garbage	Amount of space used by deleted files.
free	Amount of free space.
total	Total FLASH size.
diagnostic counters	Counts of the successes and failures for each type of FFS operation.



FFS failure counts do not necessarily mean that an error has occurred, but are also incremented if the file specified could not be found. For example attempting to delete a file which does not exist will result in the delete failures count being incremented.

See Also **ACTIVATE FLASH COMPACTION**
SHOW FLASH PHYSICAL

SHOW FLASH PHYSICAL

Syntax SHOW FLASH PHYSICAL

Description This command displays physical information about the specific type of FLASH installed in the router (Figure 1-19 on page 1-69, Table 1-14 on page 1-69).

Figure 1-19: Example output from the SHOW FLASH PHYSICAL command.

```
total size ..... 4 MBytes
device type ..... 28F008
devices ..... 4
location ..... SIMM stick
programming power ..... off
block erase time ..... 1600 milliseconds
total erase blocks ..... 64
erase block size ..... 128 kBytes
erase bit state ..... 1
page buffers ..... 0
size of page buffer ... 0 bytes
```

Table 1-14: Parameters displayed in the output of the SHOW FLASH PHYSICAL command.

Parameter	Meaning
total size	The amount of FLASH memory installed.
device type	The type of FLASH device installed.
devices	The number of FLASH devices installed.
location	The location of the FLASH memory; one of "SIMM stick" or "built in".
programming power	The state of programming power; one of "on" or "off".
block erase time	The time taken to erase an erase block.
total erase blocks	The number of erase blocks.
erase block size	The size of each erase block, in bytes.
erase bit state	The state of an erased bit.
page buffers	The number of page buffers.
size of page buffer	The size of each page buffer, in bytes.

See Also SHOW FLASH

SHOW HTTP CLIENT

Syntax SHOW HTTP CLIENT

Description This command displays the current state of the HTTP client (Figure 1-20 on page 1-70, Table 1-15 on page 1-70).

Figure 1-20: Example output from the SHOW HTTP CLIENT command.

```
HTTP Client
-----
Sessions opened ..... 0
Sessions closed ..... 0
Transmitted requests ..... 0
Received replies ..... 0
-----
```

Table 1-15: Parameters displayed in the output of the SHOW HTTP CLIENT command.

Parameter	Meaning
Sessions opened	The number of HTTP client sessions that have been started.
Sessions closed	The number of HTTP client sessions that have been closed.
Transmitted requests	The number of HTTP GET and POST requests transmitted by the client.
Received replies	The number of HTTP responses received by the client.

Examples To display the current status of the HTTP client, use the command:

```
SHOW HTTP CLIENT
```

See Also DISABLE HTTP DEBUG
DISABLE HTTP SERVER
ENABLE HTTP DEBUG
ENABLE HTTP SERVER
RESET HTTP SERVER
SHOW HTTP DEBUG
SHOW HTTP SERVER
SHOW HTTP SESSION

SHOW HTTP DEBUG

Syntax SHOW HTTP DEBUG

Description This command displays the debugging options currently enabled for the HTTP server (Figure 1-21 on page 1-70, Table 1-16 on page 1-71).

Figure 1-21: Example output from the SHOW HTTP DEBUG command.

```
Enabled Debug Modes
-----
AUTH,MSG
-----
```

Table 1-16: Parameters displayed in the output of the SHOW HTTP DEBUG command.

Parameter	Meaning
Enabled Debug Modes	The debugging modes currently enabled for the HTTP server; one or more of "NONE", "AUTH", "MSG", "SESSION" or "ALL".

Examples To display the currently enabled debugging modes for the HTTP server, use the command:

```
SHOW HTTP DEBUG
```

See Also DISABLE HTTP DEBUG
DISABLE HTTP SERVER
ENABLE HTTP DEBUG
ENABLE HTTP SERVER
RESET HTTP SERVER
SHOW HTTP CLIENT
SHOW HTTP SERVER
SHOW HTTP SESSION

SHOW HTTP SESSION

Syntax SHOW HTTP SESSION

Description This command displays TCP session information for the HTTP server (Figure 1-22 on page 1-71, Table 1-17 on page 1-72).

Figure 1-22: Example output from the SHOW HTTP SESSION command.

Session	In Use	Type	TCP State	Activations
session1	FALSE	None	-	0
session2	FALSE	None	-	0
session3	FALSE	None	-	0
session4	FALSE	None	-	0
session5	FALSE	None	-	0
session6	FALSE	None	-	0
session7	FALSE	None	-	0
session8	FALSE	None	-	0

Table 1-17: Parameters displayed in the output of the SHOW HTTP SESSION command.

Parameter	Meaning
Session	The session ID for a session. A maximum of 8 sessions can be active at any one time.
In Use	Whether or not the session is active; one of "TRUE" or "FALSE".
Type	The type of session; one of "None" (no active session), "Client" (the session is an outgoing connection from the router's HTTP client to a remote HTTP server) or "Server" (the session is an incoming connection from a client to the router's HTTP server).
TCP State	The current state of the TCP state machine; one of "FREE", "CLOSED", "LISTEN", "SYNSENT", "SYNRECEIVED", "ESTABLISHED", "FINWAIT1", "FINWAIT2", "CLOSEWAIT", "LASTACK", "CLOSING", "TIMEWAIT" or "DELETE".
Activations	The number of times this session has been activated.

Examples To display TCP session information for the HTTP server, use the command:

```
SHOW HTTP SESSION
```

See Also DISABLE HTTP DEBUG
DISABLE HTTP SERVER
ENABLE HTTP DEBUG
ENABLE HTTP SERVER
RESET HTTP SERVER
SHOW HTTP CLIENT
SHOW HTTP DEBUG
SHOW HTTP SERVER

SHOW HTTP SERVER

Syntax SHOW HTTP SERVER

Description This command displays configuration and status information for the HTTP server (Figure 1-23 on page 1-72, Table 1-18 on page 1-74).

Figure 1-23: Example output from the SHOW HTTP SERVER command.

```

HTTP Server
-----
Status ..... Enabled
Listen port ..... Open

Sessions opened ..... 0
Sessions closed ..... 0
Received requests ..... 0
Unknown requests ..... 0
Transmitted replies ..... 0
Authorisation successes .... 0
Authorisation failures .... 0
-----

```


Table 1-18: Parameters displayed in the output of the SHOW HTTP SERVER command.

Parameter	Meaning
Status	The status of the HTTP server; one of "Enabled" or "Disabled".
Listen port	Whether or not the HTTP server's TCP listen port is open; one of "Open" or "Closed".
Sessions opened	The number of HTTP server sessions that have been started.
Sessions closed	The number of HTTP server sessions that have been closed.
Received requests	The number of HTTP GET and POST requests received by the server.
Unknown requests	The number of unrecognised HTTP requests received by the server.
Transmitted replies	The number of HTTP responses transmitted by the server.
Authorisation successes	The number of successful authentication attempts received by the server.
Authorisation failures	The number of authentication failures during login attempts. Authentication failures result from users entering an invalid or no username and password when prompted by the browser.

Examples To display the current status of the HTTP server, use the command:

```
SHOW HTTP SERVER
```

See Also DISABLE HTTP DEBUG
DISABLE HTTP SERVER
ENABLE HTTP DEBUG
ENABLE HTTP SERVER
RESET HTTP SERVER
SHOW HTTP CLIENT
SHOW HTTP DEBUG
SHOW HTTP SESSION

SHOW INSTALL

Syntax SHOW INSTALL

Description This command shows the install information, which install the router is currently running and the history of checking install information at boot (Figure 1-24 on page 1-75, Table 1-19 on page 1-75).

Figure 1-24: Example output from the SHOW INSTALL command.

Install	Release	Patch	Dmp
Temporary	-	-	-
Preferred	flash:1-190b.rez	-	-
Default	EPROM (LC-1.0.0)	-	-

Current install			

Preferred	flash:1-190b.rez	-	-

Install history			

No Temporary release selected			
Preferred release selected			
Preferred release successfully installed			

Table 1-19: Parameters displayed in the output of the SHOW INSTALL command.

Parameter	Meaning
Install	The type of install; one of "Temporary", "Preferred" or "Default".
Release	The release file for the install.
Patch	<i>Not Used.</i>
Dmp	The third party Data Manipulation Program for the install. This is not present on most models and software releases.
Current install	The install currently running in the router.
Install history	A list of checks the INSTALL module carried out on the install boot. This list shows how the current install came to be selected and loaded.

See Also DELETE INSTALL
SET INSTALL

SHOW LOADER

Syntax SHOW LOADER

Description This command displays the default values for the LOADER module and, if a load is on progress, values for the current load and the progress of the current load (Figure 1-25 on page 1-76, Table 1-20 on page 1-76).

Figure 1-25: Example output from the SHOW LOADER command.

```

Loader Information
-----
Defaults:
Method..... TFTP
File ..... 28-761.rel
Destination ..... Flash
Server ..... 172.16.1.1
Proxy Port ..... Undefined
Port ..... Undefined
Delay (sec) ..... 0

Current Load:
Method..... TFTP
File ..... 28-761.rel
Destination ..... Flash
Server ..... 172.16.1.1
Proxy Port ..... Undefined
Port ..... Undefined
Delay (sec) ..... 0
Status ..... Loading
Load Level ..... 24%
Last Message ..... -
-----

```

Table 1-20: Parameters displayed in the output of the SHOW LOADER command.

Parameter	Meaning
Default Method	The default method used by the LOADER module to load files; one of "TFTP", "HTTP", "WEB", "WWW", "ZMODEM" or "NONE".
Default File	The default file name for the LOADER module.
Default Destination	The default destination for the LOADER module; "FLASH".
Default Server	The default server for the LOADER module. The IP address of the default proxy server when the method is HTTP via a proxy server.
Default Proxy Port	The default port on the proxy server for the LOADER module when the method is HTTP.
Default Port	The default port for the LOADER module if the method is ZMODEM.
Default Delay	The default delay for the LOADER module.
Current Method	The actual method being used to download the current file; one of "TFTP", "HTTP", "WEB", "WWW", "ZMODEM" or "NONE".
Current File	The actual file name for the current load.
Current Destination	The actual destination for the current load; "FLASH".
Current Server	The actual server for the current load. The actual proxy server for the current load when the method is HTTP via a proxy server.
Current Proxy Port	The actual port of the proxy server for the current load when the method is HTTP via a proxy server.
Current Port	The actual port for the current load when the method is ZMODEM.
Current Delay	The actual delay for the current load.

Table 1-20: Parameters displayed in the output of the SHOW LOADER command.

Parameter	Meaning
Status	The status of the LOADER module; one of "Idle", "Waiting", "Loading", "Load Complete" or "Load Aborted". If the SHOW LOADER command shows a Status of "Load Complete" or "Load Aborted", the next SHOW LOADER command will show a Status of "Idle" (unless another LOAD is initiated first).
Load Level	The progress of the load as a percentage of the file downloaded. This is only displayed if the LOADER Status is "Loading".
Last Message	The last error or informational message sent to the device from which the last LOAD command on page 1-42 was issued. At router boot, the Last Message is undefined and shows as "-".

See Also LOAD
 SET LOADER
 UPLOAD

SHOW MANAGER PORT

Syntax SHOW MANAGER PORT

Description This command displays the port number of the current semipermanent manager port, if any. There may be no more than one semipermanent manager port at any time. If a semipermanent manager port is defined, a message like:

```
The manager port is port 0
```

is displayed. If no semipermanent manager port is defined, the message:

```
No manager port is defined.
```

is displayed.

See Also LOGIN
 SET MANAGER PORT
 SET PORT in *Chapter 2, Interfaces*

SHOW STARTUP

Syntax SHOW STARTUP

Description This command prints the state of the bits in the router Startup Status Flag (Figure 1-26 on page 1-78). This command can be used to check the state of the router when it last started up. If a given bit signals an error then its message has an > appended to the front of it.

Figure 1-26: Example output from the SHOW STARTUP command.

```

Router Startup Status Flag is 00600040, which means:
-----
    4096k of RAM found
> Router CRASHED prior to this startup
  Battery backed RAM battery OK
  Battery backed RAM not corrupted
  Real time clock not corrupted
  Real time clock, time set
  Router software download OK
  Router vector download OK
-----

```

SHOW SYSTEM

Syntax SHOW SYSTEM

Description This command displays general system information about the router, including the hardware installed, memory, and software release loaded (Figure 1-27 on page 1-78, Table 1-21 on page 1-79). It will also display location and contact details if these have been set with the appropriate SET SYSTEM command.

Figure 1-27: Example output from the SHOW SYSTEM command.

```

Router System Status                               Time 17:10:06 Date 25-Sep-1999.
Board      ID   Bay Board Name                     Rev      Serial number
-----
Base       80   AR140(U)                               M1-0     6845218
-----
Memory -   DRAM : 4096 kB   FLASH : 4096 kB
-----
SysDescription
CentreCOM AR140(U) version 1.9.1-00 21-Feb-2000
SysContact
David Johns, ext 8331
SysLocation
Laboratory, First Floor, Head Office Building
SysName
LAB
SysUpTime
250074 ( 00:41:40 )
Software Version: 1.9.1-00 21-Feb-2000
Release Version : 1.9.1-00 21-Feb-2000
Patch Installed : NONE
Territory      : europe
Help File      : help.hlp

Boot configuration file: load.cfg (exists)
Current configuration: load.cfg
Security Mode   : Disabled

Warning (248283): No patches found.

```

Table 1-21: Parameters displayed in the output of the SHOW SYSTEM command.

Parameter	Meaning
Board	The board type; "Base".
ID	The identification number of the board.
Bay	<i>Not Used.</i>
Board Name	The descriptive name of the board.
Rev	The revision number and hardware modification level of the board.
Serial number	The serial number of the board.
DRAM	The amount of DRAM memory installed.
FLASH	The amount of FLASH memory installed.
SysDescription	A description of the product and software release.
SysContact	A string specifying a contact name or address to call for the router. This is set with the SET SYSTEM CONTACT command on page 1-54.
SysLocation	A string specifying the location of the router. This is set with the SET SYSTEM LOCATION command on page 1-55.
SysName	A string specifying the name (usually the complete IP domain name) of the router. This is set with the SET SYSTEM NAME command on page 1-55.
SysUpTime	The elapsed time, in 100ths of a second, since the last router restart.
Software Version	The patch version running on the router.
Release Version	The software release running on the router.
Patch Installed	<i>Not Used.</i>
Territory	The territory in which the router is being used; one of "australia", "china", "europe", "japan", "korea", "newzealand" or "usa". This can be set with the SET SYSTEM TERRITORY command on page 1-56.
Help File	The system help file, used by the HELP command on page 1-41 for online help. This can be set with the SET HELP command on page 1-50.
Boot configuration file	The current boot configuration file set with the SET CONFIG command on page 1-49 and whether or not the file exists (Table 1-9 on page 1-62).
Current configuration	The source of the current router configuration. This can be one of a number of items, including a configuration file name or no configuration (Table 1-9 on page 1-62).
Security Mode	Whether or not security mode is enabled; one of "Enabled" or "Disabled".
Patch files	<i>Not Used.</i>

See Also

- SET HELP
- SET SYSTEM CONTACT
- SET SYSTEM LOCATION
- SET SYSTEM NAME
- SET SYSTEM TERRITORY

SHOW TIME

Syntax SHOW TIME

Description This command displays the current router time as maintained by the real-time clock. The message displayed looks like:

```
System time is 09:18:05 on 10-Jun-1997
```

See Also SET TIME

SHOW USER

Syntax SHOW USER[=*login-name*] [CONFIGURATION]

where:

- *login-name* is a character string, 1 to 32 characters in length. Valid characters are uppercase letters (A–Z), lowercase letters (a–z), and decimal digits (0–9). The string may not contain spaces.

Description This command displays the contents of the User Authentication Database or global configuration parameters and counters for the User Authentication Facility.

For a user with MANAGER privilege, the command displays the contents of the User Authentication Database. If a login name is specified, information for the specified user is displayed. If a login name is not specified the entire database is displayed (Figure 1-28 on page 1-80, Table 1-22 on page 1-81). For a user with USER privilege, parameters are not allowed, and the user's own database record is displayed.

The CONFIGURATION parameter displays global configuration parameters and counters for the User Authentication Facility (Figure 1-29 on page 1-81, Table 1-23 on page 1-81). A login name may not be specified with the CONFIGURATION parameter.

Figure 1-28: Example output from the SHOW USER command.

```
User Authentication Database
-----
Username: dave ()
  Status: enabled   Privilege: user       Telnet: no
  Logins: 0         Fails: 0           Sent: 0           Rcvd: 0
Username: manager (Manager Account)
  Status: enabled   Privilege: manager  Telnet: yes
  Logins: 2         Fails: 1           Sent: 0           Rcvd: 0
-----
```


Table 1-22: Parameters displayed in the output of the SHOW USER command.

Parameter	Meaning
Username	The login name.
Status	The current status of the entry; one of "enabled" or "disabled".
Privilege	The privilege level for this user; one of "manager" or "user".
Telnet	Whether or not the user is permitted to use the TELNET command to telnet to a host; one of "yes" or "no".
Callback number	The ISDN phone number for this user when making a call back to a remote user.
Logins	The number of times a successful login has been made using this login name.
Fails	The number of times an incorrect password was given for this login name.
Sent	The number of octets sent by the user to the router.
Rcvd	The number of octets set to the user from the router.

Figure 1-29: Example output from the SHOW USER CONFIGURATION command.

```

User Authentication Facility configuration and counters
-----
Security parameters
login failures before lockout ..... 4 (LOGINFAIL)
lockout period ..... 20 seconds (LOCKOUTPD)
manager password failures before logoff .. 3 (MANPWDFAIL)
maximum security command interval ..... 30 seconds (SECUREDELAY)
minimum password length ..... 6 characters (MINPWDLEN)
semi-permanent manager port ..... 0
Security counters
logins 7 databaseClearTotallys 0
managerPwdChanges 0
unknownLoginNames 1
totalPwdFails 5
managerPwdFails 3
securityCmdLogoffs 1
loginLockouts 1
-----

```

Table 1-23: Parameters displayed in the output of the SHOW USER CONFIGURATION command.

Parameter	Meaning
login failures before lockout	The default number of login failures allowed by a user before the login prompt is withheld for the lockout period.
lockout period	The default lockout period, in seconds, that the login prompt will be withheld from a user after a number of consecutive login failures.
manager password failures...	The default number of successive failures a manager may make entering the login password before the session is logged off.

Table 1-23: Parameters displayed in the output of the SHOW USER CONFIGURATION command. (Continued)

Parameter	Meaning
maximum security command...	The default interval, in seconds, that may elapse between successive secure commands without the manager being prompted to re-enter the login password.
minimum password length	The default value for the minimum password length.
semi-permanent manager port	The port number of the semipermanent manager port.
logins	The total number of logins by any user to the router.
managerPwdChanges	The number of times a manager privilege level password has been changed.
unknownLoginNames	the number of attempted logins with a login name that did not exist in the database.
totalPwdFails	The total number of times an incorrect password was given for a login name that exists in the database.
managerPwdFails	The number of times a manager was challenged to give their password for a security command and they entered the incorrect password.
securityCmdLogoffs	The number of times a manager was logged off because a correct password was not entered when required to validate a security command.
loginLockouts	The number of times the login lockout period was instigated because too many unsuccessful login attempts were made.
databaseClearTotallys	The number of times the database has been cleared.

See Also ADD USER
 DELETE USER
 DISABLE USER
 ENABLE USER
 PURGE USER
 RESET USER
 SET USER

UPLOAD

Syntax UPLOAD [FILE=*filename*] [SERVER=*ipadd*] [PORT=*port*]
 [METHOD=ZMODEM]

where:

- *filename* is the name of the file to upload. This may be a full path name for the file in the syntax of the TFTP server.
- *ipadd* is an IP address in dotted decimal notation.
- *port* is the number of an asynchronous port. Ports are numbered sequentially starting with port 0.

Description This command initiates a file upload from the router using TFTP or ZMODEM. Any parameters not specified use the default values set with the SET LOADER command on page 1-51.

The FILE parameter specifies the name of the file on the router's file subsystem and should be a fully qualified file name, including the device name.

The SERVER parameter specifies the IP address of the TFTP server to which the LOADER module will attempt to upload the file. The PING command on page 6-82 of *Chapter 6, Internet Protocol (IP)* for a detailed description) can be used to verify that the router can communicate with the server via IP.

Text files can be uploaded via an asynchronous port, rather than via the TFTP protocol. The PORT parameter specifies the number of the asynchronous port via which the text file will be uploaded. If PORT is specified, METHOD must also be specified.

The METHOD parameter specifies the method to use when uploading a file via an asynchronous port. If ZMODEM is specified, the ZMODEM protocol is used to upload the file. Only text files can be uploaded using the ZMODEM protocol because release files are converted from S records by the router before storing them.



For security reasons this command will only be accepted if the user has SECURITY OFFICER privilege.

Examples To upload the file SHOW.SCP stored in FLASH memory to a TFTP server with an IP address of 172.16.8.5, use the command:

```
UPLOAD FILE=FLASH:SHOW.SCP SERVER=172.16.8.5
```

See Also LOAD
SET LOADER
SHOW FILE
SHOW LOADER

Chapter 2

Interfaces

Introduction	2-2
Naming interfaces	2-2
Ethernet	2-3
Encapsulations	2-4
Configuration	2-5
Asynchronous Interfaces	2-7
Encapsulations	2-7
Configuration	2-7
Autobauding	2-11
Displaying Interfaces	2-11
Interface Link Traps	2-12
Managing Interfaces with SNMP	2-12
Command Reference	2-13
DISABLE INTERFACE LINKTRAP	2-13
DISABLE PORT	2-14
ENABLE INTERFACE LINKTRAP	2-14
ENABLE PORT	2-15
PURGE PORT	2-15
RESET ETH	2-16
RESET ETH COUNTERS	2-16
RESET PORT	2-17
RESET PORT COUNTERS	2-17
RESET PORT HISTORY	2-18
SET INTERFACE TRAPLIMIT	2-18
SET PORT	2-19
SHOW ETH CONFIGURATION	2-22
SHOW ETH COUNTERS	2-23
SHOW ETH MACADDRESS	2-29
SHOW ETH RECEIVE	2-29
SHOW INTERFACE	2-30
SHOW PORT	2-33

Introduction

This chapter describes the Ethernet and asynchronous interfaces on the router, how the interfaces can be configured, controlled and monitored, and the encapsulations supported on each interface.

The term *interface* means one of the physical ports on the router used to connect the router to a network. All data enters and leaves the router via an interface. Asynchronous ports can also be used to connect terminals, printers and terminal ports on host computers. See *Chapter 7, Terminal Server* for information about using the asynchronous ports on the router for terminal serving functions. The Basic Rate ISDN interfaces are described in *Chapter 4, Integrated Services Digital Network (ISDN)*.

An *encapsulation* is a set of rules that specify a header for each frame of data that informs the router receiving the frame what protocol is being carried in the frame. Some interface types can be used with more than one encapsulation. It is important to know about encapsulations for two reasons. Firstly, the information can be useful in debugging network problems, if traces of the packets being transmitted or received on a particular interface can be obtained. Secondly, information about encapsulations can be used to determine whether the router can interoperate with other vendors' routers, since this depends on both routers supporting the same encapsulation(s) for a particular protocol.

The encapsulations supported for the Point-to-Point Protocol are described in detail in *Chapter 3, Point-to-Point Protocol (PPP)*.

Naming interfaces

Commands used to configure an interface, or to attach a routing module to use a particular interface, include the `INTERFACE=interface` or `OVER=interface` parameter which specifies the interface to be used, by name. Interfaces are named by concatenating the interface type with the interface instance. The interface type is an abbreviation of the full name of the interface (Table 2-1 on page 2-2).

The instance is a non-negative number. Instance numbers may be chosen arbitrarily, but in practice it is usual to assign them sequentially, starting with 0. The instance for physical interfaces (e.g. Ethernet and asynchronous interfaces) is the physical port number. Physical ports are numbered starting at 0. For logical interfaces the instance number is the module instance number specified in the `CREATE` command for that module. Table 2-2 on page 2-3 shows examples of valid interface names for the router.

Table 2-1: Router interface names and types.

Type	Description
BRI	Basic Rate ISDN interface
ETH	Ethernet interface
LAPB	X.25 LAPB interface
PORT	Asynchronous interface
PPP	Point-to-Point Protocol interface
X25T	X.25 DTE interface

Table 2-2: Examples of valid interface names.

Interface name	Description
eth0	Ethernet port 0.
port4	Asynchronous port 4.
ppp1	Point-to-Point Protocol instance 1.

Ethernet

Ethernet is a general term used to describe a particular family of interface types and encapsulations. Other common terms for this type of interface are 802.3 and CSMA/CD. Various physical media can be used to carry Ethernet, including thin and thick coaxial cable, twisted pair wires and optical fibre. All these forms of Ethernet are characterised by a few common features:

- A single medium carries all incoming and outgoing traffic.
- A number of stations may use the same medium for communicating with all other stations on the medium. All stations can see all the traffic on the medium.
- Stations wait for the medium to become free before attempting to send data on it. If more than one station attempts to send data simultaneously a collision results and the data being sent becomes invalid.
- Stations can be connected to or disconnected from the medium without disturbing the other stations on the medium.
- The order in which stations are attached to the physical medium is not important.

Ethernet runs at a speed of 10 Mbps, although the practical transmission rate is usually a lot less.

Ethernet is used to provide local area networking rather than wide area networking. The installation of Ethernet media within premises is normally the responsibility of the user of the premises, rather than the telecommunications provider.

The Ethernet interfaces on the router are specified by the IEEE 802.3 or ISO 8802-3 standards.

All routers have a 10BASET and/or an AUI connector. The AUI connector is a 15-pin connector which provides a common connector for all Ethernet media variations. Transceivers are available to connect the AUI connector to any Ethernet media, including thin wire (10BASE2) and twisted pair (10BASET). The AUI interface may also connect directly to a repeater. The AUI connector allows connection to any form of Ethernet media, albeit with a transceiver. For models with dual 10BASET/AUI connectors, only one connector may be used at any one time.



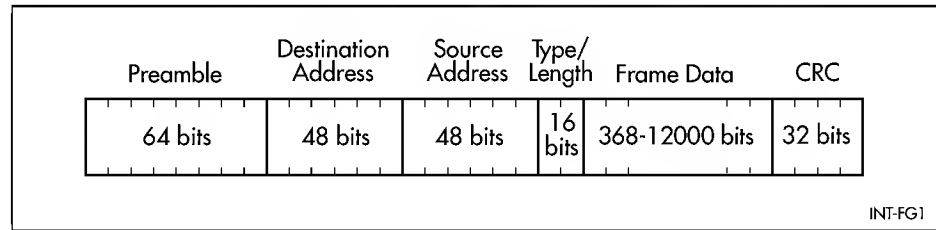
Ethernet was first defined in 1982. The original definition is generally referred to as Type 1 Ethernet and although it differs only slightly from the modern standard, it is not very common today. Subsequent standards defined Type 2 Ethernet, which was largely ratified unchanged by the IEEE as IEEE 802.3. This is the standard in use by most implementations. The router can physically support all three versions of Ethernet, and is supplied with Type 2/802.3 selected.

Encapsulations

Since Ethernet is a single wire able to be used by many stations at once, with many different protocols, encapsulation of protocol types is used to distinguish the protocols. Ethernet has been developed over a period of time, and the efforts of the standards bodies following on from the vendors that developed Ethernet, have led to different encapsulation types for Ethernet.

An Ethernet frame consists of a preamble followed by the data and terminated with a CRC (Figure 2-1 on page 2-4).

Figure 2-1: Format of an Ethernet frame.



The data begins with the station addresses of the receiver and sender of the frame. These address fields are both 6 octets long. Following the addresses is a 2-octet field, referred to here as the type/length field, that contains either a type field or a length.

The type/length field was introduced by the vendors that developed Ethernet and was used to contain a protocol type. Different values in the type field distinguished different protocols. The values that are contained in this field are administered by Xerox Corporation and vendors of network equipment may apply to reserve a type field to define vendor-specific protocols.

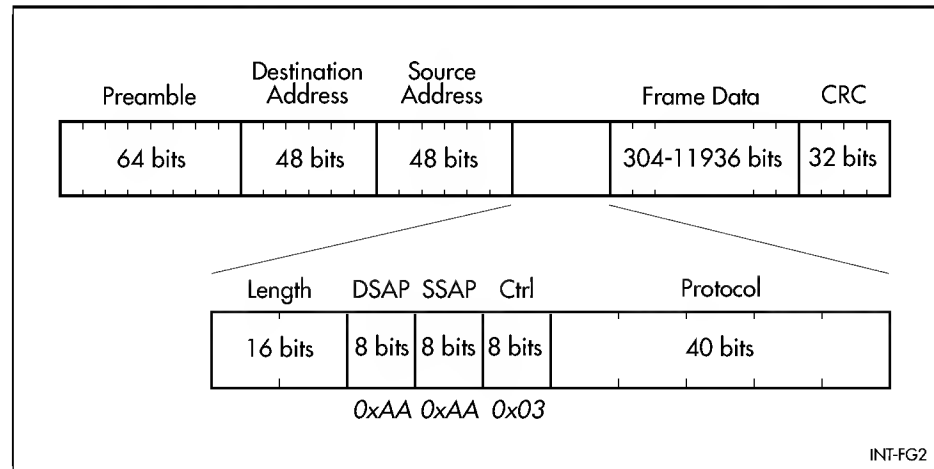
The original vendor specifications were extended by the IEEE. This body developed standards in local area networking, including Ethernet. The Ethernet addresses and type/length field appear in the IEEE standards as part of the Ethernet specific standard, IEEE 802.3. Another standard, IEEE 802.2, specifies the format of the frame after the type/length field. Since IEEE 802.2 applies to other LAN media, such as Token Ring and FDDI, the frame after the type/length field cannot contain anything specific to Ethernet. For this reason the type/length field is used to specify a length, and is, in fact, the length of the rest of the frame.

Although there may appear to be a conflict between the use of the type/length field for both a frame type and a length, in practice there is no conflict. The maximum length of an Ethernet frame (including the preamble, addresses and the type/length field) is 1514 octets, so the maximum value of the type/length field as a length is 1500 octets. Ethernet types are assigned values greater than 1500. In the early days of Ethernet, some protocol types were assigned values below 1500, but these have since become obsolete.

When the IEEE introduced its standard for Ethernet, replacing the type field with a length field, parts of the networking community decided that a way was still required to be able to specify that a particular Ethernet frame was a certain protocol type, without having to implement all of the IEEE 802.2 standard. IEEE 802.2 defines the two octets after the type/length field as *Service Access Points*, or SAPs, one for the source of the packet and one for the destination. A special SAP value (0xAA or 170 decimal) was defined to indicate that the packet containing this SAP value would use the *SubNetwork Access Protocol* (SNAP) mechanism. In IEEE 802.2, the one or two octets after the SAPs are

defined as the control field. For the SNAP format, this is defined as the single octet 0x03, used to indicate an “unnumbered information” frame. The SNAP format then defines the next 5 octets as a protocol type. Values in this field define the different protocols (Figure 2-2 on page 2-5).

Figure 2-2: Format of an Ethernet frame with SNAP encapsulation.



The router supports four encapsulation formats:

- Ethernet—type/length field used as a type.
- 802.2—use of IEEE 802.2 standard with SAPs.
- SNAP—use of the SNAP SAP.

For the correct operation of a software module the Ethernet drivers must receive packets with the appropriate encapsulation and forward them to that module. The packets are specified by an encapsulation format and a discriminator. The discriminators for each encapsulation format are listed in Table 2-3 on page 2-5.

Table 2-3: Supported Ethernet encapsulations and discriminators.

Format	Discriminator	Length (octets)
Ethernet	Ethernet type	2
802.2	Destination SAP	1
SNAP	SNAP discriminator	5

Configuration

An Ethernet interface on the router is automatically configured by the software modules when the router starts up. No user configuration of the Ethernet interfaces is required, except to enable software modules to use the interface. This is achieved by adding a software module interface and using the clause

```
INTERFACE=ETHn
```

where *n* is the number of the Ethernet interface being configured.

The modules in the router that are configured to use an Ethernet interface, and the encapsulations used on an interface, can be displayed with the command

```
SHOW ETH=n CONFIGURATION
```

where *n* is the number of the Ethernet interface.

A feature of Ethernet is the ability to send packets to more than one station at a time, using multicast addresses. The multicast addresses required by a software module are automatically entered into the list of receive addresses by that module. No action by the user is required.

Note that the list includes the broadcast address and any unicast addresses specified by the software modules that have configured to the Ethernet interface. Unicast addresses are distinguishable from multicast addresses by their first octet. The first octet of a unicast address is even, whereas for a multicast address it is odd. The broadcast address is a special multicast address that is received by all stations on an Ethernet. The router is always configured to receive broadcast packets, even if no software modules are using the interface, so the list always includes the broadcast address.

The default MAC address used by the Ethernet interface can be displayed with the command:

```
SHOW ETH [=n] MACADDRESS
```

where *n* is the number of the Ethernet interface being configured.

The addresses that the router is configured to receive can be displayed with the command:

```
SHOW ETH [=n] RECEIVE
```

where *n* is the number of the Ethernet interface being configured.

The router maintains a number of counters for each Ethernet interface. These counters are objects in three standard MIBs and the router's enterprise MIB.



For more information on MIBs, see Chapter 16, Simple Network Management Protocol (SNMP).

The counters are grouped into four categories depending on the MIB to which they belong (Table 2-4 on page 2-6).

Table 2-4: Categories of counters maintained for Ethernet interfaces.

Category	Group	MIB table	RFC
INTERFACE	Interfaces	Interfaces	1213
INTERFACE	Generic interfaces	Interface extensions	1229
DOT3STAT	Transmission	Dot 3 statistics	1398
COLLISION	Transmission	Collision statistics	1398
DIAGNOSTIC	Enterprise MIB	General	-
DIAGNOSTIC	Enterprise MIB	SONIC	-

Counters from each of these four categories are displayed using:

```
SHOW ETH [=n] COUNTERS [=category]
```

where *n* is the number of the Ethernet interface and *category* is one of the four categories of counter. If a category is not specified, all categories are displayed.

The counters in each category may be cleared to zero by the command:

```
RESET ETH [=n] COUNTERS [=category]
```

where *n* is the number of the Ethernet interface and *category* is one of the four categories of counter. If a category is not specified, all counters are cleared.



Using the RESET ETH COUNTERS command on page 2-16 to clear the counters does not clear the MIB counters themselves. Instead, the MIB counter contents are copied to offset storage locations that are subtracted from the MIB counters before being displayed by the SHOW ETH COUNTERS command on page 2-23.

The Ethernet interfaces on the router can be reset with the command:

```
RESET ETH=n
```

where *n* is the number of the Ethernet interface being reset. The Ethernet interface must be specified. A complete reset of the Ethernet interface is carried out with this command.



Any data being sent or received when the Ethernet interface is reset will be lost.

Asynchronous Interfaces

All models of the router have at least one asynchronous interface, or port. This is a standard RJ45, DB9 male or DB9 female connector wired as a DTE (*Data Terminating Equipment*) interface. The ports are identified by number, and are numbered sequentially starting from 0. The first interface is called Port 0.

All asynchronous ports use the RS-232C standard. At least four modem control lines are provided with each interface, and these would normally be used as DTR, RTS, CTS and CD.

Asynchronous ports are normally used to connect terminals or modems to the router. In general most VT100-compatible terminals will require a *crossed* (DTE-to-DTE) cable.



The term crossed refers to the fact that the data pins (TxD and RxD) on the connector at one end of the cable are connected to the opposite pins (RxD and TxD respectively) on the connector at the other end of the cable. This is necessary because both the terminal and the router have DTE interfaces.

Asynchronous ports may also be used as network interfaces.

Encapsulations

By default, no encapsulation is used on asynchronous ports. Data is transmitted and received as a clear character stream. This is appropriate for remote terminal or terminal emulation access and remote printing facilities.

Configuration

Each asynchronous port can be individually configured to suite a wide range of different terminal types. The characteristics of a port can be changed using the command:

```
SET PORT=port-number option
```

The options available for each port are listed in Table 2-5 on page 2-8. All asynchronous ports are initially configured with default values (Table 2-6 on page 2-9).

Table 2-5: Configurable parameters for asynchronous port.

Option	Description
ATTENTION	Sets the attention character used to return from a virtual terminal session to the router prompt.
DATABITS	Sets the number of data bits per character transmitted by the port.
ECHO	Enables or disables the echoing of each character entered at a terminal.
FLOW	Sets the flow control mechanism used for the port in both the receive and transmit directions. If FLOW is set to NONE the router ignores all incoming flow control characters and lead transitions. If FLOW is set to CHARACTER the router uses XON/XOFF flow control. If FLOW is set to HARDWARE the router uses the RTS/CTS lines for flow control.
INFLOW	Sets the flow control mechanism used for the port in the receive direction only. If FLOW is set to NONE the router ignores all incoming flow control characters and lead transitions. If FLOW is set to CHARACTER the router uses XON/XOFF flow control. If FLOW is set to HARDWARE the router uses the RTS/CTS lines for flow control.
HISTORY	Sets the number of commands saved for command line recall.
MAXOQLEN	Sets the maximum number of character buffers that will be permitted on the transmit queue for the port.
NAME	Assigns a text string used to identify the port, such as the name of the person whose terminal is normally connected to the port, or where the terminal is located.
OUTFLOW	Sets the flow control mechanism used for the port in the transmit direction only. If FLOW is set to NONE the router ignores all incoming flow control characters and lead transitions. If FLOW is set to CHARACTER the router uses XON/XOFF flow control. If FLOW is set to HARDWARE the router uses the RTS/CTS lines for flow control.
PAGE	Sets the number of lines of output displayed on the terminal before the router pauses and waits for the user to press a key to continue.
PARITY	Sets the parity of each character transmitted by the port.
PROMPT	Sets the prompt to a string, the default prompt, or disables the prompt.
SECURE	Controls whether a user must log in to the port before router commands will be accepted. See <i>Chapter 1, Operation</i> for information about defining users and logging in to the router.
SPEED	Sets the speed of the port, from 75 bps to 115200 bps. The terminal and port must be set to the same speed. Autobauding is also available, provided the attention character used is set to [Break]. In this mode the port will automatically adjust to the speed of the terminal that is attached, up to 19200 bps.
STOPBITS	Sets the number of stop bits per character transmitted by the port.
TYPE	Sets the terminal type to "VT100" or "DUMB". A DUMB terminal is used for printing or terminals that do not support VT100 escape sequences.

Table 2-6: Factory defaults for configurable parameters for asynchronous ports.

Option	Default setting
ATTENTION	BREAK
DATABITS	8
ECHO	ON
FLOW	HARDWARE
HISTORY	30
INFLOW	HARDWARE
MAXOQLEN	0 (Unrestricted)
NAME	Port #
OUTFLOW	HARDWARE
PAGE	22
PARITY	NONE
PROMPT	DEFAULT (CMD>)
SECURE	ON
SPEED	AUTO
STOPBITS	1
TYPE	VT100

To display the complete configuration for a particular asynchronous port use the command:

```
SHOW PORT=port-number
```

To display the complete configuration for all asynchronous ports use the command:

```
SHOW PORT=ALL
```

To display summary details for a particular asynchronous port use the command:

```
SHOW PORT=port-number SUMMARY
```

To display summary details for all asynchronous ports use the command:

```
SHOW PORT=ALL SUMMARY
```

The router maintains a separate command history list for each port, containing the last commands entered at the port. The history list can be displayed with the command:

```
SHOW PORT=port-number HISTORY
```

The router maintains a number of counters for each asynchronous port. The counters are objects in two standard MIBs and the router's enterprise MIB.



For more information about SNMP and MIBs, see Chapter 16, Simple Network Management Protocol (SNMP).

The counters are grouped into three categories depending upon the MIB to which they belong (Table 2-7 on page 2-10).

Table 2-7: Categories of counters maintained for asynchronous ports.

Category	Group	MIB table	RFC
INTERFACE	Interfaces	Interfaces	1213
RS232	Transmission	Asynchronous port	1659
DIAGNOSTIC	Enterprise MIB	Asynchronous interface	-

The MIB counters for an asynchronous port may be displayed using the command:

```
SHOW PORT [=n] COUNTERS [=category]
```

where *n* is the number of the asynchronous port and *category* is one of the three counter categories. If a category is not specified, all categories are displayed.



Objects from the general input and output signal tables (see RFC 1659) are displayed by the SHOW PORT command on page 2-33.

The counters in each category may be cleared to zero by the command:

```
RESET PORT [=n] COUNTERS [=category]
```

where *n* is the number of the asynchronous port and *category* is one of the three counter categories. If a category is not specified, all counters are cleared.



Using the RESET PORT COUNTERS command on page 2-17 to clear the counters does not clear the MIB counters themselves. Instead, the MIB counter contents are copied to offset storage locations that are subtracted from the MIB counters before being displayed by the SHOW PORT command on page 2-33.

Each asynchronous port may be enabled or disabled with the commands:

```
ENABLE PORT=n
DISABLE PORT=n
```

where *n* is the number of the asynchronous port. When an asynchronous port is disabled it will not transmit or receive any data. When the port is enabled all configuration parameters are restored to the settings in effect prior to the port being disabled. The default state of an asynchronous port is enabled.

An asynchronous port may be reset with the command:

```
RESET PORT=n
```

where *n* is the number of the asynchronous port. Any current connections are disconnected and the configuration parameters are restored from nonvolatile storage.



Any data being received or transmitted when the asynchronous port is disabled or reset will be lost.

The command history can be reset with the command:

```
RESET PORT HISTORY
```

The specific commands to change the parameters of a particular asynchronous port are given in “*Command Reference*” on page 2-13. As an example, to change the name of port 6 to “TEST” and the speed to 9600 bps, use the command:

```
SET PORT=6 NAME=TEST SPEED=9600
```



All port configuration parameters are held in nonvolatile memory, and are retained over a power cycle.

Autobauding

Asynchronous ports may be set to autobauding mode. In this mode the router will adjust the speed of the port to match the speed of the terminal attached to the port, up to a maximum speed of 19200 bps. For autobauding to work, the user should always press the [Enter] or [Return] key on the terminal several times until the router prompt appears on the screen. At this point the router has set the speed of the port. If a key other than [Enter] or [Return] is pressed while the router is setting the port speed, the speed may be incorrectly set. In this case, there will be no response from the router or “garbage” characters will appear on the terminal screen. To fix this, press [Break] two or more times, followed by [Enter] or [Return] several times.



Some terminals require the [Break] key to be held down for about a second to send a [Break] properly. Similarly, some terminals require a brief pause between multiple [Break]s.

Once the speed is set on an autobauding port, the router will not change it unless one of the following events occurs:

- The router is turned off.
- [Break] is pressed twice, in which case the router “forgets” the current speed and waits for [Enter] or [Return] to be pressed several times to set the speed again.
- The terminal is switched off. This sometimes has the effect of sending [Break]s to the router.



A port connected to a modem should always be set to a fixed speed matching that of the modem.

Displaying Interfaces

The router stores information about interfaces as objects in the Interfaces Table of MIB-II, defined in RFC 1213 “*Management Information Base for Network Management of TCP/IP-based internets: MIB-II*”. The contents of the Interfaces Table can be displayed using the command:

```
SHOW INTERFACE
```

To display more detailed information about a specific interface, use the command:

```
SHOW INTERFACE={ifIndex|interface}
```

where *ifIndex* is the index of the interface in the Interfaces Table and *interface* is the interface name. To display counters for all the interfaces, use the command:

```
SHOW INTERFACE COUNTERS
```

For a detailed description of the objects in the Interfaces Table of MIB-II, see *Appendix C, SNMP MIBs*.

Interface Link Traps

When an interface changes to or from the "Down" state an SNMP trap can be sent to any SNMP manager stations (trap hosts) which have been defined. The general operation of link traps is defined in RFC 1157, "Simple Network Management Protocol". In the typical multi-layered interface environment, each protocol layer for which an interface entry exists in the interface table can generate link up/down traps. Since interface state changes tend to propagate through the protocol layers multiple traps may be generated as the result of a single link failure. RFC 1573, "Evolution of the Interfaces Group of MIB-II", resolves this issue by providing a mechanism for enabling and disabling link trap generation on a specific interface. This allows stacked interfaces to be configured so that only one trap is sent for a link transition.

Link traps are disabled by default on the router. Link traps can be enabled or disabled on a per-interface basis, using the commands:

```
ENABLE INTERFACE={ifIndex|interface|DYNAMIC} LINKTRAP  
DISABLE INTERFACE={ifIndex|interface|DYNAMIC} LINKTRAP
```

where *ifIndex* is the value of *ifIndex* for the interface in the Interface Table and *interface* is the name of the interface. The DYNAMIC parameter handles the special case of dynamic interfaces which do not yet exist. If link traps are enabled for dynamic interfaces, a trap message is generated whenever a dynamic interface is created or destroyed. This is disabled by default.

The current settings for link traps can be displayed using the command:

```
SHOW INTERFACE={ifIndex|interface}
```

The potential exists in a large or busy network for a high volume of trap messages to be generated, especially if the network configuration involves dynamic interfaces created by ISDN calls. The maximum number of link traps generated per minute can be set for each static interface or for all dynamic interfaces, using the command:

```
SET INTERFACE={ifIndex|interface|DYNAMIC} TRAPLIMIT=1..60
```

The default is 20 trap messages per minute.

Managing Interfaces with SNMP

Router interfaces can be enabled or disabled via SNMP by setting the *ifAdminStatus* object in the *ifTable* of MIB-II MIB to 'Up(1)' or 'Down(2)' for the corresponding *ifIndex*. If it is not possible to change the status of a particular interface the router will return an SNMP error message.

The router's implementation of the *ifOperStatus* object in the *ifTable* of MIB-II MIB supports two additional values—"Unknown(4)" and "Dormant(5)" (e.g. an inactive dial-on-demand interface).



An unauthorized person, with knowledge of the appropriate SNMP community name, could bring an interface up or down. Community names act as passwords for the SNMP protocol. Care should be taken when creating an SNMP community with write access to select a secure community name and to ensure that this name is known only to authorised personnel.

Command Reference

This section describes the commands available on the router to configure and manage the Ethernet and asynchronous interfaces on the router.

Some commands require IP and SNMP to be enabled and correctly configured. See *Chapter 6, Internet Protocol (IP)* for a detailed description of the commands required to enable and configure IP. See *Chapter 16, Simple Network Management Protocol (SNMP)* for a detailed description of the commands required to enable and configure SNMP.

See "Conventions" on page xxxv of *Preface* in the front of this manual for details of the conventions used to describe command syntax. See *Appendix A, Messages* for a complete list of error messages and their meanings.

DISABLE INTERFACE LINKTRAP

Syntax `DISABLE INTERFACE={ifIndex|interface|DYNAMIC} LINKTRAP`

where:

- *ifIndex* is a decimal value specifying the entry in the interface MIB.
- *interface* is an interface name formed by concatenating an interface type and an interface instance (e.g. eth0).

Description This command disables link up/down trap generation for the specified interface. Link up/down traps are disabled by default.

The INTERFACE parameter specifies the interface for which link traps are to be disabled. The value is the interface's *ifIndex*, name, or the keyword "DYNAMIC". If DYNAMIC is specified, link trap generation for the creation and destruction of dynamic interfaces is disabled.



IP and SNMP must be enabled and correctly configured to generate traps. See Chapter 6, Internet Protocol (IP) for a detailed description of the commands required to enable and configure IP. See Chapter 16, Simple Network Management Protocol (SNMP) for a detailed description of the commands required to enable and configure SNMP.

Examples To disable link trap generation for interface ppp0, use the command:

```
DISABLE INTERFACE=PPP0 LINKTRAP
```

See Also ENABLE INTERFACE LINKTRAP
SET INTERFACE TRAPLIMIT
SHOW INTERFACE

DISABLE PORT

Syntax `DISABLE PORT=port-number`

where:

- *port-number* is the number of the port. Ports are numbered sequentially starting with port 0.

Description This command disables the specified port. The port must currently be enabled. No data will be accepted or transmitted via the specified port.

Examples To disable asynchronous port 3, use the command:

```
DISABLE PORT=3
```

See Also ENABLE PORT
PURGE PORT
RESET PORT
SET PORT
SHOW PORT

ENABLE INTERFACE LINKTRAP

Syntax `ENABLE INTERFACE={ifIndex|interface|DYNAMIC} LINKTRAP`

where:

- *ifIndex* is a decimal value specifying the entry in the interface MIB.
- *interface* is an interface name formed by concatenating an interface type and an interface instance (e.g. eth0).

Description This command enables link up/down trap generation for the specified interface. Link up/down traps are disabled by default.

The INTERFACE parameter specifies the interface for which link traps are to be enabled. The value is the interface's *ifIndex*, name, or the keyword "DYNAMIC". If DYNAMIC is specified, link trap generation for the creation and destruction of dynamic interfaces is enabled.



IP and SNMP must be enabled and correctly configured to generate traps. See Chapter 6, Internet Protocol (IP) for a detailed description of the commands required to enable and configure IP. See Chapter 16, Simple Network Management Protocol (SNMP) for a detailed description of the commands required to enable and configure SNMP.

Examples To enable link trap generation for the interface with an ifIndex of 1, use the command:

```
ENABLE INTERFACE=1 LINKTRAP
```

See Also DISABLE INTERFACE LINKTRAP
SET INTERFACE TRAPLIMIT
SHOW INTERFACE

ENABLE PORT

Syntax `ENABLE PORT=port-number`

where:

- *port-number* is the number of the port. Ports are numbered sequentially starting with port 0.

Description This command enables the specified port. The port must currently be disabled. Data will be accepted and/or transmitted via the specified port.

Examples To enable asynchronous port 3, use the command:

```
ENABLE PORT=3
```

See Also DISABLE PORT
PURGE PORT
RESET PORT
SET PORT
SHOW PORT

PURGE PORT

Syntax `PURGE PORT={port-number|ALL}`

where:

- *port-number* is the number of the port. Ports are numbered sequentially starting with port 0.

Description This command resets the specified port to the factory default configuration. If ALL is specified, all ports are reset. All current port configurations will be lost.

Examples To purge the configuration of all ports, use the command:

```
PURGE PORT=ALL
```

See Also DISABLE PORT
ENABLE PORT
RESET PORT
RESET PORT COUNTERS
RESET PORT HISTORY
SET PORT
SHOW PORT

RESET ETH

Syntax RESET ETH=*n*

where:

- *n* is the number of the Ethernet interface.

Description This command resets the specified Ethernet interface. The interface must be specified.



Any data currently being transmitted or received by the Ethernet interface will be lost. This may affect the operation of some of the protocol modules in the router.

Examples To reset Ethernet interface 0, use the command:

```
RESET ETH=0
```

See Also RESET ETH COUNTERS

RESET ETH COUNTERS

Syntax RESET ETH [=*n*] COUNTERS [= {COLLISION|DIAGNOSTIC|DOT3STAT|INTERFACE}]

where:

- *n* is the number of the Ethernet interface.

Description This command clears the status counters for an Ethernet interface. The interface number may be specified on the command line. If no interface number is specified, all interfaces have their counters cleared. If a category is specified only the counters in that category are cleared. If a category is not specified, all counters are cleared. For a description of the categories, see the description of the SHOW ETH COUNTERS command on page 2-23.



Using the RESET ETH COUNTERS command to clear the counters does not clear the MIB counters themselves. Instead, the MIB counter contents are copied to offset storage locations that are subtracted from the MIB counters before being displayed by the SHOW ETH COUNTERS command on page 2-23.

Examples To reset the interface counters for Ethernet interface 0, use the command:

```
RESET ETH=0 COUNTERS=INTERFACE
```

See Also SHOW ETH COUNTERS

RESET PORT

Syntax RESET PORT [=port-number]

where:

- *port-number* is the number of the port. Ports are numbered sequentially starting with port 0.

Description This command resets the specified port. If a port number is not specified, then the command applies to the port from which the command is issued. If a port number is specified, the command applies to the specified port. The port configuration is restored from nonvolatile storage. Any existing connections are terminated.

Examples To reset port 3, use the command:

```
RESET PORT=3
```

See Also DISABLE PORT
ENABLE PORT
PURGE PORT
RESET PORT COUNTERS
RESET PORT HISTORY
SET PORT
SHOW PORT

RESET PORT COUNTERS

Syntax RESET PORT [=port-number] COUNTERS [= {DIAGNOSTIC | INTERFACE | RS232}]

where:

- *port-number* is the number of the port. Ports are numbered sequentially starting with port 0.

Description This command resets the MIB counters for the specified asynchronous port. If a port is not specified then the counters for the port from which the command was entered are reset. If a category is specified then only the counters in that category are cleared. If a category is not entered then the counters for all categories are reset. For a description of the categories, see the description of the SHOW PORT command on page 2-33.



Using the RESET PORT COUNTERS command to clear the counters does not clear the MIB counters themselves. Instead, the MIB counter contents are copied to offset storage locations that are subtracted from the MIB counters before being displayed by the SHOW PORT command on page 2-33.



The control signal transition counters displayed by the SHOW PORT command on page 2-33 are reset along with the other counters in the RS-232 category.

Examples To reset the interface counters for port 3, use the command:

```
RESET PORT=3 COUNTERS=INTERFACE
```

See Also RESET PORT
RESET PORT HISTORY
SHOW PORT

RESET PORT HISTORY

Syntax RESET PORT [=port-number] HISTORY

where:

- *port-number* is the number of the port. Ports are numbered sequentially starting with port 0.

Description This command clears all commands from the command history for the specified port. If a port number is not specified then the command applies to the port or TTY device from which the command is issued. If a port number is specified, the command applies to the specified port.



The port history is automatically reset during the login and logoff processes.

Examples To reset the command history for the port to which the terminal is connected, use the command:

```
RESET PORT HISTORY
```

To reset the command history for port 3, use the command:

```
RESET PORT=3 HISTORY
```

See Also RESET PORT
RESET PORT COUNTERS
SHOW PORT

SET INTERFACE TRAPLIMIT

Syntax SET INTERFACE={ifIndex|interface|DYNAMIC} TRAPLIMIT=1..60

where:

- *ifIndex* is a decimal value specifying the entry in the interface MIB.
- *interface* is an interface name formed by concatenating an interface type and an interface instance (e.g. eth0).

Description This command sets the maximum number of link up/down traps generated in one minute for the specified interface. The default is 20.



IP and SNMP must be enabled and correctly configured to generate traps. See Chapter 6, Internet Protocol (IP) for a detailed description of the commands required to enable and configure IP. See Chapter 16, Simple Network Management Protocol (SNMP) for a detailed description of the commands required to enable and configure SNMP.

Examples To set the trap limit for interface ppp2 to 40, use the command:

```
SET INTERFACE=PPP2 TRAPLIMIT=40
```

See Also DISABLE INTERFACE LINKTRAP
ENABLE INTERFACE LINKTRAP
SHOW INTERFACE

SET PORT

Syntax SET PORT [=port-number] [ATTENTION={BREAK|^P|NONE}]
[DATABITS={5|6|7|8}] [ECHO={ON|OFF|YES|NO|TRUE|FALSE}]
[FLOW={CHARACTER|HARDWARE|NONE}] [HISTORY=0..99]
[INFLOW={CHARACTER|HARDWARE|NONE}]
[MAXOQLEN=0..214783647] [NAME=name]
[OUTFLOW={CHARACTER|HARDWARE|NONE}] [PAGE={4..99|OFF}]
[PARITY={EVEN|MARK|NONE|ODD|SPACE}] [PROMPT={prompt|
DEFAULT|OFF}] [SECURE={ON|OFF|YES|NO|TRUE|FALSE}]
[SPEED={AUTO|75|110|134.5|150|300|600|1200|1800|2000|
2400|4800|9600|14400|14.4K|19200|19.2K|28800|28.8K|
38400|38.4K|57600|57.6K|115200|115.2K}] [STOPBITS={1|
2}] [TYPE={DUMB|VT100}]

where:

- *port-number* is the number of the port. Ports are numbered sequentially starting with port 0.
- *name* is a character string, 1 to 15 characters in length. If the string contains spaces it must be enclosed in double quotes. The string is not case sensitive.
- *prompt* is a character string, 1 to 15 characters in length. If the string contains spaces it must be enclosed in double quotes. The string is not case sensitive.

Description This command set the characteristics of asynchronous ports. If a port is not specified, then the command applies to the port on which the command is issued. If a port number is specified, the command applies to the specified port. Multiple options may be specified in the same command.

If the SET PORT command is issued from a port with USER privilege, the port number and SECURE may **not** be specified.

For a Telnet connection only the options HISTORY, PAGE, PROMPT and TYPE may be used to alter the behaviour of the dedicated TTY device.

The change takes place immediately and the new value is stored in nonvolatile memory.

The ATTENTION parameter specifies the character used to return from an active session (e.g. a Telnet connection) to the router prompt. The default is BREAK (the [Break] key) for asynchronous ports, and ^P (the [Ctrl/P] key) for Telnet connections to the router.



If autobauding is enabled, the attention character must be set to [Break] as this is the only character that can be detected before the baud rate is established.

The DATABITS parameter sets the number of data bits per character transmitted by the port. This should match the terminal setting. The default is 8.

The ECHO parameter sets the echo mode for the port. If ECHO is set to ON characters typed following the prompt are echoed to the terminal screen. If ECHO is set to OFF, characters will not be echoed to the terminal screen but the router will still receive and process them. This option only has effect when the port is not assigned. When the port is assigned, echoing is controlled by the host. The default is ON.

The FLOW parameter sets the flow control mechanism used for the port in both the transmit and receive directions. If FLOW is set to NONE, the router ignores all incoming flow control characters or lead transitions. The router will not generate any flow control characters and the state of the hardware lines will not change. If FLOW is set to CHARACTER, the router uses XON/XOFF flow control. If FLOW is set to HARDWARE, the router uses the RTS/CTS lines for flow control. For finer control, the INFLOW and OUTFLOW parameters can be used to set different flow control mechanisms for the port in the receive and transmit directions, respectively.

The HISTORY parameter sets the number of commands saved in the command history for future recall. The minimum number is 0 and the maximum is 99. Setting the history length to zero for a port does not clear all the commands from the history. The command history is cleared with the RESET PORT HISTORY command on page 2-18. The default history length for asynchronous ports and Telnet connections is 30.

The MAXOQLEN parameter sets the maximum number of character buffers permitted on the output queue for this port. Once the queue has reached this limit no further buffers will be accepted for transmission from the higher layer. The default is 100. A value of 0 means there is no limit on the length of the output queue.

The NAME parameter assigns a name to the port, as a convenient reference to identify ports. For example, it may be set to the name of the person who normally uses the terminal connected to the port, or the location of the terminal. The default name is "Port #" where "#" is the port number. The name appears in the output of the SHOW PORT command on page 2-33.

The PAGE parameter sets the number of lines of command output displayed on the terminal screen before the router pauses and waits for the user to press a key to continue. This number may range from 4 to 99. The default is 22 for both asynchronous ports and Telnet connections. If PAGE is set to OFF, paging is disabled.

The PARITY parameter sets the parity of each character transmitted by the port. This should match the terminal setting. The default is NONE.

The PROMPT parameter sets the prompt for the port to either the default string, such as:

```
CMD>
```

or a user-specified string, or disables the prompt. It is often convenient to disable the prompt if the port is being used as a manager port or for debugging network problems, as it reduces the clutter on the terminal screen. This option only has effect when the port is not assigned. When the port is assigned, prompting is controlled by the host.

The SECURE parameter determines whether a user must log in to the port before router commands will be accepted. See *Chapter 1, Operation* for more information on logging in and defining users of the router. The default is ON for both asynchronous ports and Telnet connections.

The SPEED parameter sets the speed (baud rate) of the port. This should match the terminal setting. The attention character must be set to [Break] if autobauding is selected. The port expects to see several [Enter] or [Return] characters to determine the terminal speed setting. If another character is entered initially after the port is reset or cleared, the autobauding feature may not select the correct speed. To restart autobauding in this situation, two consecutive [Break] characters should be entered, followed by two [Enter] or [Return] characters. The default is AUTO.



Autobauding will not work with baud rates exceeding 19200 baud, the maximum for most terminals. A port connected to a modem should not be set to autobauding.



Not all speeds are supported on all router models. If an unsupported speed is specified, an error message is displayed and the command is ignored.

The STOPBITS parameter sets the number of stop bits per character transmitted by the port. This should match the terminal setting. The default is 1.

The TYPE parameter specifies the type of terminal attached to the port. If TYPE is set to VT100 the router expects the terminal to support standard VT100 escape sequences, and will use them. If TYPE is set to DUMB, the router will not use VT100 escape sequences. The DUMB option is usually only required for ports connected printers or very old terminals that do not support VT100 escape sequences. The default is VT100 for both asynchronous ports and Telnet connections.

Examples The following command configures port 17:

```
SET PORT=17 DATA=7 PA=ODD SPEED=9600 ST=1
```

Each parameter can also be set separately:

```
SET PORT=17 DATA=7
SET PORT=17 PA=ODD
SET PORT=17 SPEED=9600
SET PORT=17 ST=1
```

See Also DISABLE PORT
 ENABLE PORT
 RESET PORT
 SET TTY in *Chapter 7, Terminal Server*
 SHOW PORT
 SHOW TTY in *Chapter 7, Terminal Server*

SHOW ETH CONFIGURATION

Syntax SHOW ETH=*n* CONFIGURATION

where:

■ *n* is the number of the Ethernet interface.

Description This command lists the modules in the router that are configured to use an Ethernet interface, and the encapsulations used on an interface (Figure 2-3 on page 2-22, Table 2-8 on page 2-22).

Figure 2-3: Example output from the SHOW ETH CONFIGURATION command.

Configuration for ETH instance 0:				
Module	Protocol	Format	Discrim	MAC address
IPG	IP	Ethernet	0800	0000cd000027
IPG	ARP	Ethernet	0806	0000cd000027

Table 2-8: Parameters displayed in the output of the SHOW ETH CONFIGURATION command.

Field	Meaning
Module	The name of the software module that has configured to the Ethernet drivers to receive packets with this encapsulation.
Protocol	The name of the protocol, which is determined from the format and discriminator.
Format	The encapsulation format specified by the module.
Discrim	The discriminator specified by the module to identify which packets of the given format should be received.
MAC Address	The Media Access Control source address for which the module wishes to receive packets. This is commonly known as the Ethernet address.

See Also SHOW ETH COUNTERS
 SHOW ETH RECEIVE

SHOW ETH COUNTERS

Syntax `SHOW ETH [=n] COUNTERS [= {COLLISION | DIAGNOSTIC | DOT3STAT |
INTERFACE}]`

where:

■ *n* is the number of the Ethernet interface.

Description This command displays the MIB counters for an Ethernet interface. If the interface is not specified, the counters for all Ethernet interfaces are displayed. If a category is not specified, all counters are displayed.

If COLLISION is specified, collision statistics counters from the dot3 MIB are displayed (Figure 2-4 on page 2-23). Collision frequencies are displayed in pairs of columns, representing a histogram. The right hand column (the value axis of the histogram) is a count of individual MAC frames for which the transmission on the specified interface was accompanied by the number of per-frame media collisions specified in the left hand column (the category axis of the histogram).

If DIAGNOSTIC is specified, diagnostic counters specific to the router's Ethernet hardware are displayed. The output is divided into two parts, *device independent* counters which are the same for all router models, and *device dependent* counters which differ depending on whether the interface uses 68360 hardware (Figure 2-5 on page 2-24, Table 2-9 on page 2-24) or SONIC hardware (Figure 2-6 on page 2-24, Table 2-9 on page 2-24).

If DOT3STAT is specified, statistics counters from the dot3 MIB are displayed (Figure 2-7 on page 2-26, Table 2-10 on page 2-27).

If INTERFACE is specified, interface counters from the MIB-II MIB are displayed (Figure 2-8 on page 2-27, Table 2-11 on page 2-28).

Figure 2-4: Example output from the SHOW ETH COUNTERS=COLLISIONS command.

ETH instance 0:		1245 seconds		Last change at:		0 seconds	
dot3 MIB Collision Statistics Counters							
Collision frequencies:							
1:	11	5:	0	9:	0	13:	0
2:	9	6:	0	10:	0	14:	0
3:	3	7:	0	11:	0	15:	0
4:	0	8:	0	12:	0	16:	0

Figure 2-5: Example output from the SHOW ETH COUNTERS=DIAGNOSTIC command for 68360-based hardware.

ETH instance 0:		438 seconds	Last change at:	0 seconds
Device Independent Diagnostic Counters				
EthProtoCacheHit	463	EthProtoCacheMiss	58	
DSAPProtoCacheHit	0	DSAPProtoCacheMiss	0	
SNAPProtoCacheHit	0	SNAPProtoCacheMiss	0	
RxFIFOOverrun	0	TxFIFOUnderrun	0	
RxTooFewBuffers	0	TxTooManyFragments	0	
BusError	0	TxDescriptorAreaFull	0	
Reset	0	TxFrameTooLong	0	
LoadCAMFailure	0	TxLostInterrupt	0	
Device Dependent Diagnostic Counters				
CommandTimeout	0	TxNoPacket	0	
Command	0			

Figure 2-6: Example output from the SHOW ETH COUNTERS=DIAGNOSTIC command for SONIC-based hardware.

ETH instance 0:		131 seconds	Last change at:	0 seconds
Device Independent Diagnostic Counters				
EthProtoCacheHit	112	EthProtoCacheMiss	9	
DSAPProtoCacheHit	0	DSAPProtoCacheMiss	0	
SNAPProtoCacheHit	0	SNAPProtoCacheMiss	0	
RxFIFOOverrun	0	TxFIFOUnderrun	0	
RxTooFewBuffers	0	TxTooManyFragments	0	
BusError	0	TxDescriptorAreaFull	0	
Reset	0	TxFrameTooLong	0	
LoadCAMFailure	0	TxLostInterrupt	0	
Device Dependent Diagnostic Counters				
RxStatusError	0	TxEOLoverrun	0	
RxBuffMismatch	0	TxDescCorrupt	0	
RxBuffAreaExceeded	0	TxSpuriousBCMError	0	
RxBuffExhausted	0	TxBCMError	0	
RxDescExhausted	0	TxPacketMonitoredBad	0	

Table 2-9: Parameters displayed in the output of the SHOW ETH COUNTERS=DIAGNOSTIC command.

Counter	Meaning
EthProtoCacheHit	The number of times for an Ethernet protocol packet the Ethernet type field matched that of a protocol discriminator structure in the cache.
DSAPProtoCacheHit	The number of times for a DSAP protocol packet the DSAP field matched that of a protocol discriminator structure in the cache.
SNAPProtoCacheHit	The number of times for a SNAP protocol packet the SNAP field matched that of a protocol discriminator structure in the cache.

Table 2-9: Parameters displayed in the output of the SHOW ETH COUNTERS=DIAGNOSTIC command. (Continued)

Counter	Meaning
RxFIFOOverrun	The number of times reception of a packet failed due to a FIFO overrun.
RxTooFewBuffers	The number of times that after the reception of a packet or during recovery from a receive buffers exhausted interrupt there were insufficient free buffers to replenish the queue of buffers available for reception.
BusError	The number of times a direct memory access transfer was aborted due to a bus error.
Reset	The number of times the ETHRecover routine was called in response to a serious error.
LoadCAMFailure	The number of times a load of the CAM failed.
EthProtoCacheMiss	The number of times for an Ethernet protocol packet the Ethernet type field matched that of a protocol discriminator structure in the discriminator list but the structure was not in the cache.
DSAPProtoCacheMiss	The number of times for a DSAP protocol packet the DSAP field matched that of a protocol discriminator structure in the discriminator list but the structure was not in the cache.
SNAPProtoCacheMiss	The number of times for a SNAP protocol packet the SNAP field matched that of a protocol discriminator structure in the discriminator list but the structure was not in the cache.
TxFIFOUnderrun	The number of times the transmission of a packet failed due to a FIFO underrun.
TxTooManyFragments	The number of times a packet could not be transmitted because it contained too many fragments.
TxDescriptorAreaFull	The number of times there was insufficient room in the Transmit Descriptor Area because there were so many packets queued for transmission and not yet transmitted.
TxFrameTooLong	The number of times a frame was not transmitted because it exceeded the maximum length of an Ethernet frame.
TxLostInterrupt	The number of times the lost transmit interrupt timer timed out before a packet had been transmitted.
CommandTimeout	[68360 hardware] The number of times a command to the Ethernet hardware did not complete before the timeout timer expired.
Command	[68360 hardware] The code of the command that was to be issued when a command timeout was detected.
TxNoPacket	[68360 hardware] The number of times the Ethernet hardware reported a transmit error, but there was no packet being transmitted or the errored packet could not be identified.
RxStatusError	[SONIC hardware] The number of times the packet length or status fields of a descriptor were found to contain illegal values.

Table 2-9: Parameters displayed in the output of the SHOW ETH COUNTERS=DIAGNOSTIC command. (Continued)

Counter	Meaning
RxBuffMismatch	[SONIC hardware] The number of times when processing Receive Descriptors after a problem due to a SONIC bug has been encountered that the buffer pointed to by the Receive Resource Descriptor and the Receive Descriptor are different.
RxBuffAreaExceeded	[SONIC hardware] The number of times a packet was received that was so large it would have exceeded the receive buffer.
RxBuffExhausted	[SONIC hardware] The number of receive buffers exhausted interrupts.
RxDescExhausted	[SONIC hardware] The number of receive descriptors exhausted interrupts.
TxEOLOverrun	[SONIC hardware] The number of times the Ethernet hardware overran the end of the transmit packet list and caused a transmit error interrupt.
TxDescCorrupt	[SONIC hardware] The number of times when a transmit error occurred that the transmit descriptor was found to be corrupt.
TxSpuriousBCMError	[SONIC hardware] The number of times the Ethernet hardware reported a byte count mismatch in the packet buffer, but no error existed.
TxBCMError	[SONIC hardware] The number of times the Ethernet hardware reported a byte count mismatch in the packet buffer and such an error was found.
PacketMonitoredBad	[SONIC hardware] The number of times the SONIC receive unit monitored a transmitted packet as bad.

Figure 2-7: Example output from the SHOW ETH COUNTERS=DOT3STAT command.

ETH instance 0:		1295 seconds	Last change at:	0 seconds
dot3 Statistics MIB Counters				
Receive:		Transmit:		
InternalMacRxErrors	0	InternalMacTxErrors	0	
FrameTooLongs	0	DeferredTransmissions	5	
AlignmentErrors	0	SingleCollisionFrames	11	
FCSErrors	2	MultipleCollisionFrames	12	
Missed	0	LateCollisions	0	
UnwantedBroad	5114	ExcessiveCollisions	0	
UnwantedMulticasts	5	CarrierSenseErrors	0	
RxQueueLength	0	ExcessiveDeferrals	0	

Table 2-10: Parameters displayed in the output of the SHOW ETH COUNTERS=DOT3STAT command.

Counter	Meaning
InternalMacRxErrors	The number of frames for which reception failed due to an internal error.
FrameTooLongs	The number of frames received that exceeded the maximum permitted frame size.
AlignmentErrors	The number of received frames with alignment and FCS errors.
FCSErrors	The number of received frames with FCS but not alignment errors.
Missed	The number of packets not received due to: (1) lack of memory resources to buffer the packet, (2) a FIFO overrun, (3) receiver was disabled.
UnwantedBroad	The number of broadcast frames received on this interface for a protocol not configured to any module.
UnwantedMulticasts	The number of multicast frames received on this interface for a protocol not configured to any module.
RxQueueLength	The length of the queue of receive packets between the interrupt routine and the idle level receive packet processing routine.
InternalMacTxErrors	The number of frames not transmitted due to internal error.
DeferredTransmissions	The number of frames delayed by busy medium.
SingleCollisionFrames	The number of successfully transmitted frames that were inhibited by exactly one collision.
MultipleCollisionFrames	The number of successfully transmitted frames that were inhibited by more than one collision.
LateCollisions	The number of times that a collision is detected later than 512 bit times into the transmission of a packet.
ExcessiveCollisions	The number of frames not transmitted due to excessive collisions.
CarrierSenseErrors	The number of times that carrier sense was lost or never asserted during the transmission of a frame.
ExcessiveDeferrals	The number of times transmission of a packet was aborted due to excessive deferrals.

Figure 2-8: Example output from the SHOW ETH COUNTERS=INTERFACE command.

ETH instance 0:		1239 seconds	Last change at:	0 seconds
Interface MIB Counters				
Receive:		Transmit:		
ifInOctets	2357609	ifOutOctets	872296	
ifInUcastPkts	2588	ifOutUcastPkts	3985	
ifInNUcastPkts	420	ifOutNUcastPkts	2	
ifExtnsMulticastsRxOKs	5	ifExtnsMulticastsTxOKs	0	
ifExtnsBroadcastsRxOKs	5308	ifExtnsBroadcastsTxOKs	2	
ifInDiscards	0	ifOutDiscards	0	
ifInErrors	2	ifOutErrors	0	
ifInUnknownProtos	4888	ifOutQLen	1	

Table 2-11: Parameters displayed in the output of the SHOW ETH COUNTERS=INTERFACE command.

Counter	Meaning
ifLastChange	The value of sysUpTime at the time the interface entered its current operational state.
ifInOctets	The number of octets received on this interface.
ifInUcastPkts	The number of unicast packets delivered to a higher-layer protocol.
ifInNUcastPkts	The number of non-unicast packets delivered to a higher-layer protocol.
ifExtnsMulticastsReceivedOKs	The number of frames successfully received for a multicast address other than a broadcast address.
ifExtnsBroadcastsReceivedOKs	The number of frames successfully received for a broadcast address other than a multicast address.
ifInDiscards	The number of inbound packets discarded though no errors had been detected to preventing them from being deliverable to higher-layer protocol.
ifInErrors	The number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol.
ifInUnknownProtos	The number of packets which were discarded because they were for an unconfigured protocol.
ifOutOctets	The number of octets transmitted, including framing.
ifOutUcastPkts	The number of unicast packets transmitted or discarded.
ifOutNUcastPkts	The number of non-unicast packets transmitted or discarded.
ifExtnsMulticastsTransmittedOKs	The number of frames successfully transmitted to a multicast address other than a broadcast address.
ifExtnsBroadcastsTransmittedOKs	The number of frames successfully transmitted to a broadcast address other than a multicast address.
ifOutDiscards	The number of packets discarded though no errors had been detected preventing their being transmitted.
ifOutErrors	The number of packets not transmitted because of errors.
ifOutQLen	The length of the output packet queue.

See Also RESET ETH COUNTERS
SHOW ETH CONFIGURATION
SHOW ETH RECEIVE

SHOW ETH MACADDRESS

Syntax `SHOW ETH [=n] MACADDRESS`

where:

- `n` is the number of the Ethernet interface.

Description This command displays the default MAC address for the specified Ethernet interface. If the interface is not specified, the default MAC addresses for all Ethernet interfaces are displayed (Figure 2-9 on page 2-29).

Figure 2-9: Example output from the SHOW ETH MACADDRESS command.

```
MAC address for ETH instance 0:

Address
-----
00-00-cd-00-0d-0e
-----
```

See Also `SHOW ETH CONFIGURATION`
 `SHOW ETH COUNTERS`
 `SHOW ETH RECEIVE`

SHOW ETH RECEIVE

Syntax `SHOW ETH [=n] RECEIVE`

where:

- `n` is the number of the Ethernet interface.

Description This command displays the multicast addresses that an Ethernet interface has been configured to receive. If the interface is not specified, the multicast addresses for all Ethernet interfaces are displayed (Figure 2-10 on page 2-30).

Note that the list includes the broadcast address and any unicast addresses specified by the software modules that have configured to the Ethernet interface. Unicast addresses are distinguishable from multicast addresses by their first octet. The first octet of a unicast address is even, whereas for a multicast address it is odd. The broadcast address is a special multicast address that is received by all stations on an Ethernet. The router is always configured to receive broadcast packets, even if no software modules are using the interface, so the list always includes the broadcast address.

In Figure 2-10 on page 2-30, the first entry is the default MAC address assigned to the router. The last entry is the broadcast address.

Figure 2-10: Example output from the SHOW ETH RECEIVE command.

```
Receive addresses for ETH instance 0:
```

```
Address
-----
00-00-cd-00-0d-0e
ff-ff-ff-ff-ff-ff
-----
```

See Also SHOW ETH CONFIGURATION
SHOW ETH COUNTERS

SHOW INTERFACE

Syntax SHOW INTERFACE [= { *ifIndex* | *interface* }] [COUNTERS]

where:

- *ifIndex* is a decimal value specifying the entry in the interface MIB.
- *interface* is an interface name formed by concatenating an interface type and an interface instance (e.g. eth0).

Description This command displays the contents of the interface MIB. If an interface is not specified, summary information for all interfaces is displayed (Figure 2-11 on page 2-30, Table 2-12 on page 2-30). If an interface is specified detailed information for the specified interface is displayed including the counters for that interface (Figure 2-12 on page 2-31, Table 2-13 on page 2-31).

The COUNTERS parameter displays interface counters for all interfaces (Figure 2-13 on page 2-32 Table 2-14 on page 2-32).

Figure 2-11: Example output from the SHOW INTERFACE command.

Interfaces		sysUpTime:		03:02:35
DynamicLinkTraps.....Disabled				
TrapLimit.....20				
ifIndex	Interface	ifAdminStatus	ifOperStatus	ifLastChange

1	eth0	Up	Up	00:00:02
2	bri0	Up	Down	01:06:04

Table 2-12: Parameters displayed in the output of the SHOW INTERFACE command.

Parameter	Meaning
sysUpTime	The elapsed time since the last router restart.
DynamicLinkTraps	Whether or not link traps have been enabled for dynamic interfaces; one of "Enabled" or "Disabled".

Table 2-12: Parameters displayed in the output of the SHOW INTERFACE command. (Continued)

Parameter	Meaning
TrapLimit	The maximum number of link up/down traps that will be generated in one minute, for dynamic interfaces.
ifIndex	The index of the interface in the interface table.
Interface	The name of the interface.
ifAdminStatus	The administratively-set (configured) state of the interface; one of "Up", "Down" or "Testing".
ifOperStatus	The current operational state of the interface; one of "Up", "Down", "Testing", "Unknown" or "Dormant".
ifLastChange	The value of <i>sysUpTime</i> at the time the interface entered its current operational state.

Figure 2-12: Example output from the SHOW INTERFACE command for a specific interface.

```

Interface..... eth0
  ifIndex..... 1
  ifMTU..... 1500
  ifSpeed..... 10000000
  ifAdminStatus..... Up
  ifOperStatus..... Up
  ifLinkUpDownTrapEnable... Disabled
  TrapLimit..... 20

Interface Counters

  ifInOctets ..... 21484      ifOutOctets ..... 13775
  ifInUcastPkts ..... 165      ifOutUcastPkts ..... 134
  ifInNUcastPkts ..... 19      ifOutNUcastPkts ..... 0
  ifInDiscards ..... 0         ifOutDiscards ..... 0
  ifInErrors ..... 0           ifOutErrors ..... 0
  ifInUnknownProtos ..... 30

```

Table 2-13: Parameters displayed in the output of the SHOW INTERFACE command for a specific interface.

Parameter	Meaning
Interface	The name of the interface.
ifIndex	The index of the interface in the interface table.
ifMTU	The size, in octets, of the largest packet that can be transmitted on the interface.
ifSpeed	An estimate of the interface's current bandwidth, in bits per second.
ifAdminStatus	The administratively-set (configured) state of the interface; one of "Up", "Down" or "Testing".
ifOperStatus	The current operational state of the interface; one of "Up", "Down", "Testing", "Unknown" or "Dormant".
ifLinkUpDownTrapEnable	Whether or not link traps have been enabled for this interfaces; one of "Enabled" or "Disabled".
TrapLimit	The maximum number of link up/down traps that will be generated in one minute, for dynamic interfaces.

Table 2-13: Parameters displayed in the output of the SHOW INTERFACE command for a specific interface. (Continued)

Parameter	Meaning
Interface Counters	Counters for the interface.
ifInOctets	The number of octets (bytes) received by the interface.
ifInUcastPkts	The number of unicast packets received by the interface.
ifInNUcastPkts	The number of multicast packets received by the interface.
ifInDiscards	The number of packets discarded by the interface.
ifInErrors	The number of packets received with errors by the interface.
ifUnknownProtos	The number of packets received by the interface but discarded because their protocol is unsupported.
ifOutOctets	The number of bytes transmitted by the interface.
ifOutUcastPkts	The number of unicast packets transmitted by the interface.
ifOutNUcastPkts	The number of multicasts transmitted by the interface.
ifOutDiscards	The number of output packets discarded by the interface.
ifOutErrors	The number of packets that should have been transmitted but were not transmitted because of errors.

Figure 2-13: Example output from the SHOW INTERFACE COUNTERS command.

```

Interface Counters

Interface: eth0
  ifInOctets ..... 22852          ifOutOctets ..... 15565
  ifInUcastPkts ..... 184          ifOutUcastPkts ..... 148
  ifInNUcastPkts ..... 19          ifOutNUcastPkts ..... 0
  ifInDiscards ..... 0             ifOutDiscards ..... 0
  ifInErrors ..... 0               ifOutErrors ..... 0
  ifInUnknownProtos ..... 30

Interface: ISDN Basic Rate Interface
  ifInOctets ..... 0              ifOutOctets ..... 0
  ifInUcastPkts ..... 0            ifOutUcastPkts ..... 0
  ifInNUcastPkts ..... 0            ifOutNUcastPkts ..... 0
  ifInDiscards ..... 0            ifOutDiscards ..... 0
  ifInErrors ..... 0              ifOutErrors ..... 0
  ifInUnknownProtos ..... 0

```

Table 2-14: Parameters displayed in the output of the SHOW INTERFACE COUNTERS command.

Parameter	Meaning
Interface	The name of the interface.
ifInOctets	The number of octets (bytes) received by the interface.
ifInUcastPkts	The number of unicast packets received by the interface.
ifInNUcastPkts	The number of multicast packets received by the interface.
ifInDiscards	The number of packets discarded by the interface.
ifInErrors	The number of packets received with errors by the interface.
ifUnknownProtos	The number of packets received by the interface but discarded because their protocol is unsupported.

Table 2-14: Parameters displayed in the output of the SHOW INTERFACE COUNTERS command. (Continued)

Parameter	Meaning
ifOutOctets	The number of bytes transmitted by the interface.
ifOutUcastPkts	The number of unicast packets transmitted by the interface.
ifOutNUcastPkts	The number of multicasts transmitted by the interface.
ifOutDiscards	The number of output packets discarded by the interface.
ifOutErrors	The number of packets that should have been transmitted but were not transmitted because of errors.

Examples To display the general state of all interfaces, use the command:

```
SHOW INTERFACE
```

See Also DISABLE INTERFACE LINKTRAP
ENABLE INTERFACE LINKTRAP
SET INTERFACE TRAPLIMIT

SHOW PORT

Syntax SHOW PORT [=*port-number*|ALL] [{COUNTERS [= {DIAGNOSTIC |
INTERFACE | RS232 }] | HISTORY | SUMMARY }]

where:

- *port-number* is the number of the port. Ports are numbered sequentially starting with port 0.

Description This command displays configuration information for one or more ports. If a port number is specified then the information for the specified port is displayed. If a port number is not specified, information for the port from which the command was issued is displayed. If ALL is specified then information for all the ports on the router is displayed. If the command is issued from a port with USER privilege the port number may not be specified and the information displayed is for the port from where the command was issued.

If no parameters are specified, then full configuration information for the specified port or ports is displayed (Figure 2-14 on page 2-34, Table 2-15 on page 2-35).

The COUNTER parameter displays counters from the specified categories (Figure 2-15 on page 2-37, Table 2-16 on page 2-37). If a category is not specified then counters from all categories are displayed. If DIAGNOSTIC is specified then counters from the asynchronous interface table of the router enterprise MIB are displayed. If INTERFACE is specified then interface counters from the interfaces MIB are displayed. Interface MIB counters only exist for ports that are in use as network interfaces. If RS-232 is specified then counters from the asynchronous port table of the RS-232like hardware devices MIB are displayed. The COUNTERS parameter may also not be specified from a port with USER privilege.

The HISTORY parameter displays the command history for the specified port or ports (Figure 2-16 on page 2-38). The command history can also be displayed by pressing [Ctrl/C]. After displaying the command history the router prompts for a command number from the list. The user can enter a number and press [Enter] or [Return] to select a command, or just press [Enter] or [Return] to return to the prompt. If a valid command number is entered then the command is displayed at the prompt ready for editing and execution.

The SUMMARY parameter displays a one-line summary for the specified port or ports (Figure 2-17 on page 2-38, Table 2-17 on page 2-38).

Figure 2-14: Example output from the SHOW PORT command.

```

PORT 2 : 0000070953 seconds   Last change at: 0000009023 seconds

PORT information
Name ..... Port 2
Status ..... enabled
Mode ..... PPP
Data rate ..... 38400
Parity ..... none
Data bits ..... 8
Stop bits ..... 1
Test mode ..... no
In flow state (mode) ..... on (Hardware)
Out flow state (mode) ..... off (Hardware)
Autobaud mode ..... disabled
Max tx queue length ..... 100
TX queue length ..... 0
Transmit frame ..... none
RX queue length ..... 0

TTY information
Instance ..... 18
Login Name .....
Description ..... Port 2
Secure ..... yes
Connections to .....
Current connection ..... none
In flow state ..... on
Out flow state ..... on
Attached module ..... Terminal server
Attached module instance .. 2
Type ..... VT100
Prompt ..... login
Echo ..... yes
Attention ..... break
Manager ..... no
Edit mode ..... insert
History length ..... 20
Page size ..... 22

```

Table 2-15: Parameters displayed in the output of the SHOW PORT command.

Parameter	Meaning
Name	The name of the port.
Status	The status of the port (enabled or disabled).
Mode	The mode of operation for the port. This will be "Ten" for terminal server ports (characters bundled every tenth of a second).
Data rate	The baud rate for the port. By default this is "Auto" to signify that the port will autobaud.
Parity	The parity setting for the port.
Data bits	The number of data bits in each transmitted character and the number expected in each received character.
Stop bits	The number of stop bits transmitted after each character and the number expected after each received character.
Test mode	Whether or not the interface is in a test mode; one of "yes" or "no".
In flow state (mode)	The flow control state and mode for the incoming data path. The flow control state may be "on" or "off", indicating whether or not the port is able to receive characters. The mode may be "none" (no flow control), "hardware" (RTS/CTS flow control), or "XON/XOFF" (XON/XOFF flow control).
Out flow state (mode)	The flow control state and mode for the outgoing data path. See "In flow state" for a description. The mode is the same for both directions.
Autobaud mode	The autobauding mode (one of "enabled" or "disabled"), and, if it is "enabled", the current state of the autobauding process; one of "searching" (the port is trying to determine the baud rate of the terminal) or "found" (the baud rate has been set).
Max tx queue length	The maximum number of character buffers that will be permitted on the transmit queue for the port. This parameter only affects a port used as a network interface.
Tx queue length	The length of the queue of character buffers that are waiting to be transmitted to the port.
Transmit frame	The address of the current frame being transmitted by the port, or "none" if no frame is currently being transmitted.
Rx queue length	The length of the queue of character buffers that are waiting to be passed up from the port to higher layers.
Instance	The instance number for the TTY device dedicated to this port.
Login name	The login name of the user logged in to this port (if any).
Description	The name assigned to the port.
Secure	Whether or not the port is secure; one of "yes" or "no".
Connections to	A list of TTY devices (if any) to which this port TTY is linked for the purpose of providing multiple sessions.
Current connection	The instance number of the TTY that this port TTY is currently connected to, or "none" if there is no active connection.

**Table 2-15: Parameters displayed in the output of the SHOW PORT command.
(Continued)**

Parameter	Meaning
In flow state	The input flow control state for the TTY dedicated to this port.
Out flow state	The output flow control state for the TTY dedicated to this port.
Attached module	The module that owns the port, by default this will be "Terminal server".
Attached module instance	The instance of the module that owns the port.
Type	The terminal type setting for the port; one of "dumb" or "VT100".
Prompt	The type of prompt given on this port; one of "default", "off", "login", "password", "confirm", "encapsulation", or a user-defined string.
Echo	Whether or not the port echoes input characters; one of "yes" or "no".
Attention	The attention character for this port; one of "none", "break" or "char". For an asynchronous port the default attention character is "break".
Manager	Whether or not the port has MANAGER privilege; one of "yes" or "no".
Edit mode	The edit mode for the port; one of "?", "insert" or "overstrike". The default is "insert".
History length	The maximum number of commands that will be held in the command history for this port. The default is 30.
Page mode/length	The number of lines of command output the router will display before pausing and waiting for the user to press a key, or "off" if page mode is disabled for this port. The default is 22.

Figure 2-15: Example output from the SHOW PORT COUNTERS command.

Port 1: 0000014132 seconds Last change at: 0000000000 seconds			
RS-232 MIB Counters			
Receive:			
ParityErrs	0		
FramingErrs	0		
OverrunErrs	0		
Diagnostic Counters			
Receive:		Transmit:	
inCharacters	690025	outCharacters	689828
inBuffers	13513	outBuffers	13526
fcsErrors	0	droppedBuffers	0
Interface MIB Counters			
Receive:		Transmit:	
ifInOctets	690025	ifOutOctets	689828
ifInUcastPkts	13513	ifOutUcastPkts	13526
ifInNUcastPkts	0	ifOutNUcastPkts	0
ifInDiscards	0	ifOutDiscards	0
ifInErrors	0	ifOutErrors	0
ifInUnknownProtos	0	ifOutQLen	0

Table 2-16: Parameters displayed in the output of the SHOW PORT COUNTERS command.

Parameter	Meaning
ParityErrs	The number of characters received with a parity error.
FramingErrs	The number of characters received with a framing error.
OverrunErrs	The number of characters lost due to an overrun error.
inCharacters	The total number of characters received.
inBuffers	The number of character buffers transferred to a higher layer.
fcsErrors	The number frames received with a frame check sequence error.
outCharacters	The total number of characters transmitted.
outBuffers	The number of character buffers transmitted for a higher layer.
droppedBuffers	The number of character buffers discarded because the output queue had reached its maximum allowed length.
ifInOctets	The number of octets received on this interface.
ifInUcastPkts	The number of unicast packets delivered to a higher-layer protocol.
ifInNUcastPkts	The number of non-unicast packets delivered to a higher-layer protocol.
ifInDiscards	The number of inbound packets discarded though no errors had been detected to preventing them from being deliverable to higher-layer protocol.
ifInErrors	The number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol.
ifInUnknownProtos	The number of packets which were discarded because they were for an unconfigured protocol.
ifOutOctets	The number of octets transmitted, including framing.

Table 2-16: Parameters displayed in the output of the SHOW PORT COUNTERS command. (Continued)

Parameter	Meaning
ifOutUcastPkts	The number of unicast packets transmitted or discarded.
ifOutNUcastPkts	The number of non-unicast packets transmitted or discarded.
ifOutDiscards	The number of packets discarded though no errors had been detected preventing their being transmitted.
ifOutErrors	The number of packets not transmitted because of errors.
ifOutQLen	The length of the output packet queue.

Figure 2-16: Example output from the SHOW PORT HISTORY command.

```

1  sh po cou
2  sh po sum
3  sh po hist
4  sh po cou
5  login manager
6  sh tty
7  sh po=1 cou
8  sh syn cou=int
9  sh po=1 cou

Enter command number>

```

Figure 2-17: Example output from the SHOW PORT SUMMARY command.

```

Port Name           Module Mode   Data Format Attn Secur Mgr
-----
001 Port 1          TTY      TEN      19200,N,8,1 brk  yes   no
-----

```

Table 2-17: Parameters displayed in the output of the SHOW PORT SUMMARY command.

Parameter	Meaning
Port	The number of the port.
Name	The name assigned to the port.
Module	The module that owns the port.
Mode	The mode of operation for the port.
Data Format	The baud rate, parity, number of data bits and number of stop bits configured for the port.
Attn	The attention character for the port; one of "-", "brk" or "chr".
Secur	Whether or not the port is secure; one of "yes" or "no".
Mgr	Whether or not the port has MANAGER privilege; one of "yes" or "no".

Examples To show the configuration for port 1, use the command:

```
SHOW PORT=1
```

To show all the counters for port 1 enter:

```
SHOW PORT=1 COUNTERS
```

To see the command history for the port to which the terminal is connected enter:

```
SHOW PORT HISTORY
```

To obtain an abbreviated display for port 1 enter:

```
SHOW PORT=1 SUMMARY
```

See Also DISABLE PORT
ENABLE PORT
RESET PORT
RESET PORT COUNTERS
RESET PORT HISTORY
SET MANAGER PORT in *Chapter 1, Operation*
SET PORT
SET TTY in *Chapter 7, Terminal Server*
SHOW TTY in *Chapter 7, Terminal Server*

Chapter 3

Point-to-Point Protocol (PPP)

Introduction	3-3
The Point-to-Point Protocol	3-3
Encapsulation	3-3
Control Protocols	3-4
LCP Options	3-5
Link Quality Management	3-6
Multilink PPP	3-6
Bandwidth Allocation Protocol	3-7
Dial-On-Demand	3-8
Bandwidth on Demand	3-8
PPP Over Ethernet	3-8
Templates	3-9
PPP Callback	3-11
Magic Number	3-12
Authentication Protocols	3-12
Password Authentication Protocol (PAP)	3-12
Challenge-Handshake Authentication Protocol (CHAP)	3-13
Configuring Authentication	3-13
Assigning IP Addresses	3-15
PPP Link Management	3-16
Debugging PPP Links	3-17
Support for PPP	3-17
Configuration Examples	3-20
Configuring a PPP link	3-20
Multilink Aggregation	3-22
Dial on Demand Links	3-24
Link Quality Monitoring	3-24
Compression	3-25
Bandwidth on Demand	3-25
Command Reference	3-27
ACTIVATE PPP	3-27
ADD PPP	3-28
CREATE PPP	3-31
CREATE PPP TEMPLATE	3-36
DELETE PPP	3-41
DESTROY PPP	3-41
DESTROY PPP TEMPLATE	3-42
DISABLE PPP	3-42
DISABLE PPP DEBUG	3-43
DISABLE PPP TEMPLATE DEBUG	3-43
ENABLE PPP	3-44

ENABLE PPP DEBUG	3-45
ENABLE PPP TEMPLATE DEBUG	3-46
PURGE PPP	3-47
RESET PPP	3-48
SET PPP	3-49
SET PPP TEMPLATE	3-54
SHOW PPP	3-58
SHOW PPP CONFIG	3-59
SHOW PPP COUNT	3-65
SHOW PPP DEBUG	3-75
SHOW PPP IDLETIMER	3-76
SHOW PPP LIMITS	3-77
SHOW PPP MULTILINK	3-78
SHOW PPP NAMESERVER	3-80
SHOW PPP TEMPLATE	3-80
SHOW PPP TXSTATUS	3-84

Introduction

This chapter describes the main features of the Point-to-Point Protocol (PPP), support for the Point-to-Point Protocol on the router, and how to configure network interfaces on the router to use the Point-to-Point Protocol.

The Point-to-Point Protocol was developed by the Internet Engineering Task Force (IETF) as a means of transmitting data for more than one network protocol over the same point-to-point serial link in a standard, vendor-independent way. The Point-to-Point Protocol provides mechanisms for transmitting data over ISDN calls, groups of TDM slots, and Ethernet.

The Point-to-Point Protocol

The Point-to-Point Protocol consists of three main components:

- A method for encapsulating datagrams over serial links.
- A Link Control Protocol (LCP) for establishing, configuring, and testing the data-link connection.
- A family of Network Control Protocols (NCPs) for establishing and configuring different network-layer protocols.

The mechanism that PPP uses to carry network traffic is to open a link with a short exchange of packets. Once the link is open, network traffic is carried with very little overhead. Frames are sent as unnumbered information frames, so no data link acknowledgement is required and no retransmissions are carried out. Once the link is established, PPP acts as a straight data pipe for protocols.

Encapsulation

The Point-to-Point Protocol is, at the lowest level, an example of the HDLC protocol, with the following features:

- Data comes in frames, delimited by special characters called flags.
- When a frame is not being sent, the sender transmits flags continually. This means that there is constant activity on any line that is running properly.
- The first four bytes of a PPP frame comprise a 1 octet address field which is always set to 0xFF, a 1 octet control field which is always set to 0x03 ("unnumbered information") and a 2-octet protocol field.
- The data that follows the address and control fields is interpreted by the device receiving the frame depending on the encapsulation type.

A Link Control Protocol (LCP) exists to bring up the PPP link before any other protocols can begin transmission. Each protocol carried over PPP has an associated Network Control Protocol (NCP) that negotiates options for the protocol and brings up the link for that protocol (Table 3-1 on page 3-4).

Table 3-1: Supported Network protocols and Network Control Protocols for the Point-to-Point Protocol.

Protocol	PPP Type (hexadecimal)
LCP	0xC021
IP	0x0021
IPCP	0x8021
TCP/IP Comp	0x002D
TCP/IP Uncomp	0x002F
Multilink	0x003D
Individual Link Compression	0x00FB
ILCCP	0x80FB
Compression	0x00FD
CCP	0x80FD
Link Quality Report	0xC025
Password Authentication Protocol (PAP)	0xC023
Challenge-Handshake Authentication Protocol (CHAP)	0xC223



The TCP/IP Comp and TCP/IP Uncomp protocols provide direct support for Van Jacobson's header compression. For more information on Van Jacobson's header compression see Chapter 6, Internet Protocol (IP).

Control Protocols

Control protocols are protocols run by PPP between the two stations at either end of a link to allow the link to be used to carry a particular type of traffic. The Link Control Protocol (LCP) must run before any other control protocol in order to allow the link to be used at all.

The local and remote stations negotiate the configuration options to be used on the link. A *configure request packet* is sent first containing configuration options. The remote station responds with a packet confirming that the options are okay, suggesting different options or rejecting the options. This exchange takes place in both directions and when a station has sent and received an acknowledge packet the link is declared open.

Once the link has been opened by the LCP, any authentication that is required is performed. When authentication has been completed successfully, or if no authentication is required, then a Network Control Protocol (NCP) is run for each network layer protocol using the link. The NCPs operate in a similar way to the LCP, negotiating configuration options specific to the network layer protocol. No NCPs can use the PPP link until the LCP has opened the link, and no data packets can be exchanged unless the appropriate NCP is open.

Control protocols consist of states, events and packets. Events cause the state of a link to change (Table 3-2 on page 3-5). Two important events are OPEN and CLOSE. They can be caused either by a management command or internally, for example, when the router powers up. An OPEN event causes the control protocol to try to establish a link and a CLOSE event terminates a link. Other events are the hardware becoming available (UP) or unavailable (DOWN), timeouts, and the arrival of packets.

Table 3-2: States for control protocols of the Point-to-Point Protocol.

State	Meaning
INITIAL	Startup state; no OPEN event has occurred and the hardware is DOWN.
STARTING	An OPEN event has occurred and the hardware is DOWN.
CLOSED	The hardware is UP and no OPEN event has occurred.
STOPPED	The hardware is UP and a DOWN or TIMEOUT event has occurred.
CLOSING	The link has been UP and a CLOSE event has occurred; trying to close link.
STOPPING	The link has been OPEN and the remote station is trying to CLOSE the link.
REQ SENT	A configure request has been sent; waiting for a reply.
ACK RCVD	A configure request has been sent, and an acknowledge received.
ACK SENT	A configure request has been received, and an acknowledge sent.
OPENED	An acknowledge has been sent and received.

The state of a PPP link (LCP) and the NCPs running on that link can be displayed with the command:

```
SHOW PPP
```

LCP Options

The LCP will attempt to negotiate the following options:

- Maximum Receive Unit (MRU).
- Endpoint Discriminator.
- Link Discriminator, as defined in RFC 2125.
- Authentication Protocol.
- Link Quality Reporting (LQR).
- Magic Number.
- Maximum Received Reconstructed Unit (MRRU).

All other options are set to the default values specified in the relevant RFC.

Endpoint Discriminator Option

The Endpoint Discriminator Option is defined in RFC 1990 and is required for PPP to form multilink bundles from dynamic PPP calls. The Endpoint Discriminator provides a mechanism for identifying the physical location of the peer at the remote end of a PPP link. When two or more dynamic PPP calls are made from the same peer, with the same authentication information, they can be bundled together to form a multilink interface if they have the same Endpoint Discriminator. The router uses its MAC address to identify itself.

If an Endpoint Discriminator is received during LCP negotiation on a newly activated link in a static PPP interface with more than one link, and that Endpoint Discriminator value is different from the Endpoint Discriminators received during negotiation on the other active links in the interface, then the new link with the invalid Endpoint Discriminator will be deactivated.

Link Discriminator Option

The Link Discriminator Option is defined in RFC 2125 and is required for the operation of BAP. During LCP negotiation it is used to declare a unique identifier for the link over which the negotiation is occurring. This unique identifier is used by BAP to differentiate the various links in a multilink bundle.

Link Quality Management

Link quality management is used to determine the quality of a PPP link. A *Link Quality Report* (LQR) packet is transmitted down the link by the router at regular intervals. This LQR packet contains information which is used to determine how many packets are being lost on the link. The interval between transmissions of LQR packets is determined by the LQR timer value obtained from the peer during the negotiation of the LQR LCP option. This timer value, which defines how often the peer expects to see an LQR packet, is configured at the peer using the commands:

```
CREATE PPP=ppp-interface OVER=physical-interface LQR=time  
SET PPP=ppp-interface OVER=physical-interface LQR=time
```

If an LQR packet is not seen by the peer within twice the configured timer value the link is deemed to have failed and is reset.

Each LQR packet also contains the magic number determined during the LCP negotiation process. If the magic number in an incoming LQR packet is the same as the local magic number then the link is deemed to be in loopback mode and is reset.

Multilink PPP

PPP provides a mechanism for combining a number of PPP links into a single bundle of links, whose bandwidth is the sum of the bandwidths of the individual links. This mechanism is known as multilink PPP (MP) and is described in RFC 1990.

When a packet is transmitted over a multilink bundle it is encapsulated by a multilink header which includes information to allow the packets sent over the links in the bundle to be sequenced. This gives the multilink bundle the same properties as a single PPP link. This encapsulation also includes information that allows large packets to be fragmented, spreading the data across a number of links and giving better packet throughput in some circumstances.

When a packet is about to be transmitted across a PPP multilink bundle, a decision is made as to which link to use to transmit the packet. If all link speeds in the multilink bundle are the same, and packets are being transmitted at a rate so that each packet has been transmitted before the next packet arrives for transmission, a round-robin scheme is used to choose between links. If there is a choice between two or more equally desirable links the packet will be sent on the link that was least recently used. Rotating traffic in this way prevents links from remaining idle for long periods of time and reduces the number of null fragments that must be transmitted during idle periods.

Both static and dynamic PPP interfaces can be multilinked. The Endpoint Discriminator LCP option ("*Endpoint Discriminator Option*" on page 3-5)

enables a single dynamic PPP interface to accept and bundle more than one call. If two or more dynamic PPP calls are made from the same peer with the same authentication information, they will be bundled together to form a multilink interface.



Van Jacobson's TCP/IP header compression should not be enabled on a multilink PPP interface.

Bandwidth Allocation Protocol

The Bandwidth Allocation Protocol (BAP), defined in RFC 2125, provides a mechanism for two PPP peers to manage the bandwidth available to the protocols using a multilink PPP bundle by negotiating gracefully to add and remove links from the multilink bundle. The negotiation process allows each peer to choose the algorithm used to determine when to add or remove links in the multilink bundle.

The Bandwidth Allocation Control Protocol (BACP), defined in RFC 2125, is a standard PPP NCP protocol used to negotiate the use of BAP on a multilink PPP interface. BACP is negotiated once per multilink bundle. If BACP is negotiated on any of the links in a multilink bundle, it is opened for all of the links in the bundle. BACP must be successfully negotiated before BAP can be used.

The Favoured Peer Option is the only option defined for BACP and is used to determine which peer is favoured in the event that both peers simultaneously transmit the same BAP request. Each peer negotiates a 4-octet magic number, which is successfully negotiated when the two magic numbers are different. The favoured peer is the peer with the lowest magic number.

After BACP reaches the opened state, either peer can request that another link be added to the bundle by sending a BAP *Call-Request* or *Callback-Request* packet. A *Call-Request* packet is sent if the peer wishes to originate the call for the new link, and a *Callback-Request* packet is sent if the peer wishes its remote peer to originate the call for the new link.

A peer can also request that a link be dropped from the bundle. A BAP *Link-Drop-Query-Request* packet is sent to the remote peer to negotiate dropping a link. The link will remain active as long as the remote peer considers the link necessary and rejects the *Link-Drop-Query-Request*. A peer can force the dropping of a link without negotiation by sending an LCP *Terminate-Request* packet on the link.

BAP can be configured when a PPP interface is created, using the command:

```
CREATE PPP=ppp-interface OVER=physical-interface BAP={ON|OFF}
      BAPMODE={CALL|CALLBACK}
```

or by modifying an existing PPP interface, using the command:

```
SET PPP=ppp-interface BAP={ON|OFF} BAPMODE={CALL|CALLBACK}
```

By default, BAP is enabled ("ON"). If BAP is disabled, PPP will use the UPRATE, UPTIME, DOWNRATE and DOWNTIME parameters to manage bandwidth on demand (see "Bandwidth on Demand" on page 3-8).

Dial-On-Demand

A PPP interface that uses an ISDN call as the physical interface can be configured for dial-on-demand. The call is activated only when there is traffic to transmit over the PPP interface. The call is disconnected when the link has been idle for a period of time specified by the IDLE parameter of the CREATE PPP command on page 3-31 and the SET PPP command on page 3-49. If the IDLE parameter is set to OFF, the dial-on-demand feature is disabled. The configured and current timer values can be displayed using the command:

```
SHOW PPP IDLETIMER
```

Bandwidth on Demand

A PPP interface over a number of ISDN channels can be configured to provide bandwidth on demand.

One application of bandwidth on demand is the use of ISDN calls to provide additional bandwidth to a leased line during peak load periods. This application is best suited to a network connection that has a fairly constant load most of the time, but is overloaded during peak periods. A leased line with sufficient capacity to handle the normal loading is supplemented by a connection to an ISDN service. This avoids the high cost of a leased line capable of handling peak loads but which is under-utilised most of the time.

A second application of bandwidth on demand is the use of multiple ISDN connections, instead of a leased line, to provide the bandwidth required at any one time. This application is best suited to a network connection that has a variable and irregular load.

To trigger the addition and removal of channels, the total utilisation of the PPP interface as a percentage of the maximum bandwidth of the PPP interface is measured every second. Each time the utilisation remains above the threshold specified by the UPRATE parameter for a time longer than that specified by the UPTIME parameter, a new ISDN call is made increasing the bandwidth of the PPP interface. When each new link is added the total utilisation of the interface will decrease. However, this decrease will only be momentary if the rate of utilisation is increasing. When the rate of utilisation decreases again, each time the utilisation drops below the threshold specified by the DOWNRATE parameter for a time longer than that specified by DOWNTIME parameter, an ISDN call will be disconnected and the total bandwidth of the PPP interface will be decreased.

PPP Over Ethernet

PPP over Ethernet, defined in RFC 2516 "*A Method of Transmitting PPP Over Ethernet*", provides the ability to connect a network of hosts over a single bridging access device to a remote *Access Concentrator*. An Access Concentrator may offer multiple services. A PPP over Ethernet link is a point-to-point connection between a host and a single service on an Access Concentrator. Typically, the bridging access device is a DSL or cable modem to which local hosts are connected via Ethernet, and the remote Access Concentrator is a server at an ISP. PPP over Ethernet enables multiple hosts at a remote site to

share the same access device, while providing the access control and billing functionality of dial-up PPP connections.

The router behaves as a host, as defined in RFC 2516, creating PPP links over Ethernet to services on remote Access Concentrators.

PPP Over Ethernet has two distinct stages. In the *Discovery Stage*, the host discovers all the available Access Concentrators that offer the required service and then selects one. The host broadcasts an *Initiation* packet specifying the name of the service to which the host wants to connect. Access Concentrators which support the requested service respond with *Offer* packets which specify the Access Concentrator's unicast Ethernet address. The host then selects an Access Concentrator and sends a *Discovery Request* packet specifying the name of the service to which the host wants to connect. The Access Concentrator responds with a *Discovery Session Confirmation* packet. When the Discovery Stage is complete the host and the selected Access Concentrator have all the information they need to create the point-to-point connection over Ethernet. In the *Session Stage* the host and the Access Concentrator exchange PPP packets.

The CREATE PPP and ADD PPP commands enable a PPP over Ethernet service to be specified as the physical interface for a PPP interface:

```
CREATE PPP=ppp-interface OVER=physical-interface
[other-ppp-options]...

ADD PPP=ppp-interface OVER=physical-interface
[other-ppp-options]...
```

where *ppp-interface* is the PPP interface number and *physical-interface* is the name of the physical interface in the format *ETHn-servicename*. Service names may be up to 18 characters in length.

Templates

Dynamic PPP interfaces are created in response to a request from a lower layer (e.g. ISDN) to create a new PPP interface. PPP templates enable the full range of configuration options available on static PPP interfaces to be applied to dynamic PPP interfaces.

A template is a blueprint for the configuration of dynamic PPP interfaces, specifying any of the parameters that may be configured on a static PPP interface. A new template is created using the command:

```
CREATE PPP TEMPLATE=template [COPY=template]
[AUTHENTICATION={CHAP|EITHER|PAP|NONE}] [BAP={ON|OFF}]
[BAPMODE={CALL|CALLBACK}] [CBDELAY=1..100]
[CBMODE={ACCEPT|OFF|REQUEST}] [CBNUMBER=e164number]
[CBOPERATION={E164NUMBER|USERAUTH}]
[COMPALGORITHM=STACLZS] [COMPRESSION={ON|OFF|LINK}]
[DEBUGMAXBYTES=16..256] [DESCRIPTION=description]
[ECHO={ON|OFF|period}] [FRAGMENT={ON|OFF}]
[FRAGOVERHEAD=0..100] [IDLE={ON|OFF|time}]
[INDATALIMIT={NONE|1..65535}] [IPREQUEST={ON|OFF}]
[LOGIN=USER] [LQR={ON|OFF|time}] [MAGIC={ON|OFF}]
[MAXLINKS=1..64] [MULTILINK={ON|OFF}] [NULLFRAGTIMER=time]
[ONLINELIMIT={NONE|1..65535}] [OUTDATALIMIT={NONE|
1..65535}] [PASSWORD=password] [RESTART=time]
[STACHECK={LCB|SEQUENCE}] [TOTALDATALIMIT={NONE|
1..65535}] [USERNAME=username]
```

An existing template can be modified or deleted using the commands:

```
SET PPP TEMPLATE=template [AUTHENTICATION={CHAP|EITHER|PAP|
NONE}] [BAP={ON|OFF}] [BAPMODE={CALL|CALLBACK}]
[CBDELAY=1..100] [CBMODE={ACCEPT|OFF|REQUEST}]
[CBNUMBER=e164number] [CBOperation={E164NUMBER|USERAUTH}]
[COMPALGORITHM=STACLS] [COMPRESSION={ON|OFF|LINK}]
[DEBUGMAXBYTES=16..256] [DESCRIPTION=description]
[ECHO={ON|OFF|period}] [FRAGMENT={ON|OFF}]
[FRAGOVERHEAD=0..100] [IDLE={ON|OFF|time}]
[INDATALIMIT={NONE|1..65535}] [IPREQUEST={ON|OFF}]
[LOGIN=USER] [LQR={ON|OFF|time}] [MAGIC={ON|OFF}]
[MAXLINKS=1..64] [MULTILINK={ON|OFF}] [NULLFRAGTIMER=time]
[ONLINELIMIT={NONE|1..65535}] [OUTDATALIMIT={NONE|
1..65535}] [PASSWORD=password] [RESTART=time]
[STACHECK={LCB|SEQUENCE}] [TOTALDATALIMIT={NONE|
1..65535}] [USERNAME=username]

DESTROY PPP TEMPLATE=template
```

The list of currently defined templates, including the default template, can be displayed using the command:

```
SHOW PPP TEMPLATE
```

The configuration of a specific template can be displayed using the command:

```
SHOW PPP TEMPLATE=template
```

Once a template has been created, it can be associated with an ISDN call using the commands:

```
ADD ISDN CALL=name NUMBER=number PRECEDENCE={IN|OUT}
PPPTEMPLATE=template

SET ISDN CALL=name PPPTEMPLATE=template
```

When the lower layer activates a call that creates a dynamic PPP interface, PPP uses the associated template to create and configure the dynamic PPP interface. If a template has not been specifically associated with a dynamic PPP interface the default template will be used.



The router will not allow configuration templates to be associated with TDM interfaces.

The full range of PPP debugging options can be enabled or disabled on a PPP template, using the commands:

```
ENABLE PPP TEMPLATE=template DEBUG={ALL|AUTH|BAPPKT|BAPSTATE|
CALLBACK|DEMAND|ENCO|LCP|NCP|PKT|UTILISATION} [, ...]
[PORT=port-number] [TIMEOUT={NONE|1..4000000000}]
[NUMPKTS={CONT|1..4000000000}]

DISABLE PPP TEMPLATE=template DEBUG={ALL|AUTH|BAPPKT|
BAPSTATE|CALLBACK|DEMAND|ENCO|LCP|NCP|PKT|
UTILISATION} [, ...]
```

Table 3-4 on page 3-45 lists the debugging options and their meanings. Any dynamic PPP interface created from a template that has debugging enabled will display the requested debug information. Debugging will cease when the dynamic PPP interface is destroyed.

PPP Callback

The PPP callback feature allows a PPP link to be configured to accept callback requests or to make callback requests. A callback request is made during the LCP negotiation using the LCP callback option which is defined in RFC 1570 as an LCP extension. This option contains a callback operation that specifies how the peer determines the number to use when making the call back and contains a message field whose contents are dependent on the operation being used.

A PPP link is configured to make callback requests with the command:

```
SET PPP=ppp-interface OVER=physical-interface CBMODE=REQUEST
```

A PPP link is configured to accept callback requests with the command:

```
SET PPP=ppp-interface OVER=physical-interface CBMODE=ACCEPT
```

Two types of callback request operations are supported by the router—user authentication and E.164 number. The user authentication callback operation specifies that the number to call back is contained in the User Authentication Database and is obtained during authentication just prior to the call being brought down. To configure user authentication callback, use the command:

```
SET PPP=ppp-interface OVER=physical-interface  
CBOPERATION=USERAUTH
```

The E.164 number operation specifies that the callback number is contained in the message field of the callback option. When the E.164 number operation is configured for requesting a callback, the E.164 number must also be provided. To configure E.164 number callback, use the command:

```
SET PPP=ppp-interface OVER=physical-interface  
CBOPERATION=E164NUMBER CBNUMBER=e164number
```

The CBOPERATION parameter is only valid when the callback mode is set to request callback.

A PPP link that is configured to accept callback requests must also be configured to request authentication. This is necessary to prevent unauthorised peers from requesting a callback.

When a callback request is accepted, and authentication succeeds, the call is brought down and a call is made back to the peer making the request. If authentication fails the link is brought down and no call back is made. In order to cope with any variable delays in bringing down the ISDN call due to any differences in ISDN switches, a delay between bringing down the call and attempting to make the call back can be configured. The units of this delay are tenths of seconds and it is configured using the command:

```
SET PPP=ppp-interface CBDELAY=1..100
```

The CBDELAY parameter is only valid when the callback mode is set to accept callback requests.

Dynamic PPP interfaces can support PPP callback provided the dynamic PPP interface is created using a PPP template in which PPP callback has been configured, for example:

```
CREATE PPP TEMPLATE=9 DESCRIPTION="Dynamic PPP interface with  
callback" CBOPERATION=USERAUTH CBMODE=REQUEST CBDELAY=10
```



The PPP callback feature is currently only supported on PPP links over ISDN calls.

Magic Number

The magic number option is used for loopback detection. A PPP interface that is looped back will not enter the OPENED state if the magic number option is enabled. The magic number option is enabled with the **MAGIC** parameter of the **ADD PPP** command on page 3-28, the **CREATE PPP** command on page 3-31 and the **SET PPP** command on page 3-49.

Authentication Protocols

The PPP Link Control Protocol (LCP) is responsible for establishing, configuring and testing data link connections. Part of the process of configuring a link is the negotiation of various options, including an authentication protocol, which is performed before allowing Network Layer protocols to transmit data over the link. The local device performing the authentication is known as the *authenticator*. The device being authenticated is known as the *peer*.

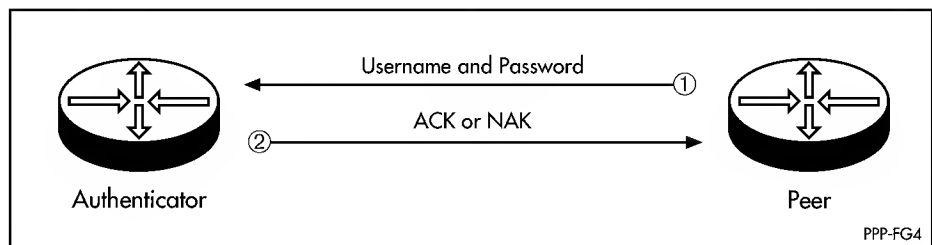
The router supports two authentication protocols: the *Password Authentication Protocol (PAP)* and the *Challenge-Handshake Authentication Protocol (CHAP)*. These protocols are primarily intended for use by PCs and hosts connecting to the router via ISDN calls.

After the PPP link has been established (the Link Establishment phase), an optional Authentication phase will take place before proceeding to the Network-Layer Protocol phase if authentication has been negotiated by the router at either end of the link.

Password Authentication Protocol (PAP)

The Password Authentication Protocol (PAP) is a relatively simple authentication protocol that allows a peer to establish its identity by repeatedly transmitting a user name/password pair to an authenticator until the authenticator acknowledges the peer or terminates the link. The peer requesting authentication controls the process; the authenticator simply responds to requests (Figure 3-1 on page 3-12).

Figure 3-1: The Password Authentication Protocol (PAP) authentication process.



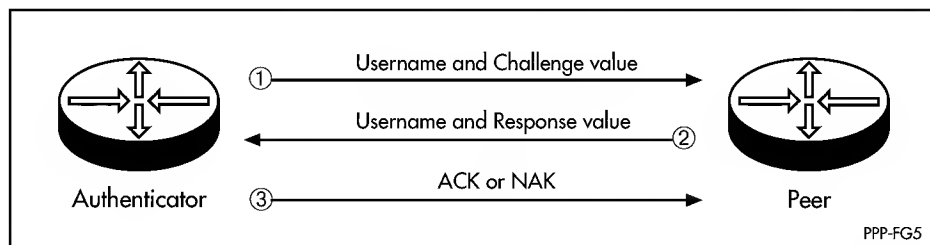
Transmitted passwords are not encrypted, and since the peer always uses the same user name/password pair there is no protection from playback or repeated trial-and-error attacks. PAP provides a similar level of security to a normal remote login.

Challenge-Handshake Authentication Protocol (CHAP)

The Challenge-Handshake Authentication Protocol (CHAP) is a more robust protocol which provides for both authentication during the Link Establishment phase and periodic verification during the Network-Layer Protocol phase.

CHAP is controlled by the authenticator, which sends a *challenge* message containing an identifier and a unique challenge value to the peer. The peer responds with a user name and value calculated by applying a *one-way hash function* (MD5) to a string created by concatenating the identifier, the password for the user name and the challenge value. The authenticator compares the response against its own computation of the function, using the user name to look up the password in the User Authentication Database. If the values match, the authentication is acknowledged, otherwise the link is terminated (Figure 3-2 on page 3-13).

Figure 3-2: The Challenge Handshake Authentication Protocol (CHAP) authentication process.



The challenge is repeated periodically during the Network-Layer Protocol phase to ensure that there has been no change to the link. Each challenge uses a different identifier and challenge value. The identifier value changes in a predictable way (typically the value of a regularly incremented counter), but the challenge value is a unique and random value. The repeated challenges and changing identifier and challenge values provide protection against both playback and trial-and-error attacks. The uniqueness and random nature of the challenge value prevents an attacker from tricking a peer into responding to a challenge then using the response to masquerade as the peer to an authenticator.

CHAP relies on the password being known to both the authenticator and the peer, although the password is not transmitted over the link.

In the case of two routers communicating over a network link, the user name is the user name set for the PPP interface with the USERNAME parameter of the CREATE PPP command on page 3-31 and the SET PPP command on page 3-49, or if this is not set, the router's system name is set with the SET SYSTEM NAME command on page 1-55 of *Chapter 1, Operation*. The password is the password set for the PPP interface with the PASSWORD parameter of the CREATE PPP and SET PPP commands.

Configuring Authentication

The router can be configured to provide authentication in one of three modes:

- As a peer in a one-way authentication scheme.
- As an authenticator in a one-way authentication scheme.
- As both a peer and an authenticator in a two-way authentication scheme.

Configuring the Router as a Peer

The router can be configured as the peer in a one-way authentication scheme. When the router makes a call to a remote device (e.g. another router configured as an authenticator), it supplies a username and password to the remote device. The remote device will determine whether or not the username and password are valid, and accept or reject the connection. This is the most common PPP authentication configuration. A typical example is a router configured to dial into a remote ISP. The ISP provides clients with a username and password. The ISP's connection server (typically a router) expects the clients router to supply the username and password when it makes a call to the ISP.

The router's username and password (supplied by the ISP) are set using the USERNAME and PASSWORD parameters in the CREATE PPP command on page 3-31 and the SET PPP command on page 3-49:

```
CREATE PPP=ppp-interface OVER=physical-interface
      USERNAME=username PASSWORD=password
SET PPP=ppp-interface USERNAME=username PASSWORD=password
```

If a username is not set using the PPP commands the peer router's system name is used as the username. The system name is set using the SET SYSTEM NAME command on page 1-55 of *Chapter 1, Operation*. The password can only be set using the PPP commands.

The router will respond to either PAP or CHAP authentication requests by supplying the configured username and password.



Neither PAP or CHAP have been explicitly configured in this example, so the router will not request authentication from remote devices during LCP negotiation. This is appropriate for the example above. Most ISPs configure their connection servers only to request authentication, not to respond to authentication requests. If the client router is configured to request authentication but the ISP's connection server is not configured to respond to authentication requests (as is typical), the ISP's connection server will refuse the connection from the client router and all connection attempts will fail.

Configuring the Router as an Authenticator

The router can be configured as the authenticator in a one-way authentication scheme. When the router receives a call from a remote device (e.g. another router configured as a peer), it requests authentication from the remote device. The remote device will supply a username and password which the router will validate before accepting or rejecting the connection. A typical example is an ISP configuring a router to accept dial-in connections from users or other remote devices (routers). The ISP's router will request authentication from the user. The user is expected to reply with the username and password supplied by the ISP when the user signed up for the service.

To configure the router as an authenticator, use the AUTHENTICATION parameter of the CREATE PPP command on page 3-31, the ADD PPP command on page 3-28 and the SET PPP command on page 3-49 to specify whether or not authentication is required, and if so, which authentication protocol to use:

```
CREATE PPP=ppp-interface OVER=physical-interface
      AUTHENTICATION={NONE | PAP | CHAP | EITHER}
ADD PPP=ppp-interface OVER=physical-interface
      AUTHENTICATION={NONE | PAP | CHAP | EITHER}
SET PPP=ppp-interface OVER=physical-interface
      AUTHENTICATION={NONE | PAP | CHAP | EITHER}
```

The AUTHENTICATION parameter must be set to PAP, CHAP or EITHER and a username and password must be entered in the User Authentication Database for each user PC (or peer) that is allowed to dial in to the router. Each peer that wants to connect to the PPP interface on the authenticator must have a username and password configured that matches one of those stored in the User Authentication Database in the authenticator.

The EITHER option uses the PPP option negotiation process to request CHAP authentication. If the peer supports CHAP, CHAP will be used. If the peer does not support CHAP, but does support PAP, PAP will be used. If the peer supports neither authentication protocol then the link is terminated.

Configuring the Router as an Authenticator and a Peer

The router can be configured as both an authenticator and a peer in a two-way authentication scheme. When the router makes a call to a remote device (e.g. another router configured as an authenticator), it supplies a username and password to the remote device. The remote device will determine whether or not the username and password are valid, and accept or reject the connection. When the router receives a call from a remote device (e.g. another router configured as a peer), it requests authentication from the remote device. The remote device will supply a username and password which the router will validate before accepting or rejecting the connection. A typical example is two routers configured to communicate via ISDN. Each router is configured as both a peer and an authenticator, as described in *"Configuring the Router as a Peer"* on page 3-14 and *"Configuring the Router as an Authenticator"* on page 3-14. Either router can make a call to the other router, supplying a username and password as authentication, or accept a call from the other router and request authentication.

The AUTHMODE parameter can be used to control when authentication will be requested. If AUTHMODE is set to INOUT authentication will be requested for both incoming and outgoing calls. Some devices will not accept calls if the calling router also requests authentication from the called router. In this case AUTHMODE can be set to IN so that only incoming calls result in authentication requests.

Assigning IP Addresses

The router supports multiple methods for assigning IP addresses to dynamic dial-in calls. The following procedure is used to select the IP address assigned to a dial-in call:

1. If the PPP interface has been added to the IP module using the ADD IP INTERFACE command on page 6-53 of *Chapter 6, Internet Protocol (IP)*, then the IP address, network mask and MTU will be as defined for that IP interface.
2. If the user is authenticated by the User Authentication Database and an IP address and MTU are associated with the user's login name, then they are used for the interface.
3. If the PPP call has an IP pool set, and the request to the IP pool is successful, then that IP address is used. See *"IP Address Pools"* on page 6-32 of *Chapter 6, Internet Protocol (IP)* for more information about creating IP address pools.
4. If all of the above steps fail to provide the necessary information then a message is displayed and the call is dropped.

See “IP Address Pools” on page 6-32 of *Chapter 6, Internet Protocol (IP)* for more information about creating IP address pools.

To associate an IP address pool with a PPP interface so that connections using that interface will use IP addresses from the IP address pool, use either of the commands:

```
CREATE PPP=ppp-interface OVER=physical-interface
      IPPOOL=pool-name [other-ppp-options...]

SET PPP=ppp-interface IPPOOL=pool-name [other-ppp-options...]
```

To disassociate an IP address pool from a PPP interface so that connections using that interface will no longer use IP addresses from the IP address pool, use the command:

```
SET PPP=ppp-interface IPPOOL=NONE
```

To associate an IP address pool with a PPP template so that dynamic PPP interfaces created using the PPP template will use IP addresses from the IP address pool, use either of the commands:

```
CREATE PPP TEMPLATE=template IPPOOL=pool-name
      [other-template-options...]

SET PPP TEMPLATE=template IPPOOL=pool-name
      [other-template-options...]
```

To disassociate an IP address pool from a PPP template so that dynamic PPP interfaces created using the PPP template will no longer use IP addresses from the IP address pool, use the command:

```
SET PPP TEMPLATE=template IPPOOL=NONE
```

PPP Link Management

Link management allows users to limit the connection time and data throughput on a PPP interface to thresholds they choose. For example, a user with an Internet connection via an Internet Service Provider (ISP) can limit the connection time or the amount of data transmitted over the PPP interface. By resetting the PPP link counters at the beginning of each billing period, they can keep their ISP bills within chosen limits.

Counters record cumulative up-time and input and output data throughput for each PPP link. The user can set thresholds for these parameters. If any of the thresholds are exceeded, the PPP link is closed. The link can not be reopened until the counters are reset, or the threshold limits are increased or disabled.

The router writes the accumulated counters to FLASH memory every five minutes and every time the PPP link is closed. If the router is restarted, the counters are restored from FLASH memory to their previous values.

To create a new PPP interface with uptime and data throughput thresholds, use the command:

```
CREATE PPP=ppp-interface [OVER=physical-interface]
      [ONLINELIMIT={NONE|1..65535}]
      [INDATALIMIT={NONE|1..65535}]
      [OUTDATALIMIT={NONE|1..65535}]
      [TOTALDATALIMIT={NONE|1..65535}] [other-options]...
```

To specify up-time and data throughput thresholds for an existing PPP interface, use the command:

```
SET PPP=ppp-interface [ONLINELIMIT={NONE|1..65535}]
[INDATALIMIT={NONE|1..65535}]
[OUTDATALIMIT={NONE|1..65535}]
[TOTALDATALIMIT={NONE|1..65535}]
```

Similarly, a PPP template with thresholds is created with the command:

```
CREATE PPP TEMPLATE=template [ONLINELIMIT={NONE|1..65535}]
[INDATALIMIT={NONE|1..65535}]
[OUTDATALIMIT={NONE|1..65535}]
[TOTALDATALIMIT={NONE|1..65535}] [other-options]...
```

Thresholds can be set on an existing PPP template using the command:

```
SET PPP TEMPLATE=template [ONLINELIMIT={NONE|1..65535}]
[INDATALIMIT={NONE|1..65535}]
[OUTDATALIMIT={NONE|1..65535}]
[TOTALDATALIMIT={NONE|1..65535}]
```

To reset the counters that record cumulative uptime and data throughput for a PPP interface to zero (0), use the command:

```
RESET PPP=ppp-interface [COUNTERS]
[LINKCOUNTERS={ONLINE|INDATA|OUTDATA|TOTALDATA|ALL}]
```

To display the cumulative counters, thresholds and remaining time or data throughput available on the interface, use the command:

```
SHOW PPP[=ppp-interface] LIMITS
```

Debugging PPP Links

Debugging can be enabled or disabled on a PPP interface, using the commands:

```
ENABLE PPP=ppp-interface DEBUG={ALL|AUTH|BAPPKT|BAPSTATE|
CALLBACK|DEMAND|ENCO|LCP|NCP|PKT|UTILISATION} [, ...]
[PORT=port-number] [TIMEOUT={NONE|1..4000000000}]
[NUMPKTS={CONT|1..4000000000}]
DISABLE PPP=ppp-interface DEBUG={ALL|AUTH|BAPPKT|BAPSTATE|
CALLBACK|DEMAND|ENCO|LCP|NCP|PKT|UTILISATION} [, ...]
```

Table 3-4 on page 3-45 lists the debugging options and their meanings. Output is sent to the specified asynchronous port or the terminal from which the command was entered.

Support for PPP

The router supports PPP over ISDN calls (see *Chapter 4, Integrated Services Digital Network (ISDN)*), MIOX calls (see *Chapter 5, X.25*), TDM groups (see *Chapter 11, Time Division Multiplexing (TDM)*), and Ethernet separately and as members of a multilink bundle. PPP can be used on the router to carry IP and compressed data.

A PPP interface is created and associated with a “physical interface” (an ISDN call, a MIOX circuit, a TDM group, or Ethernet)

```
CREATE PPP=interface OVER=physical-interface
```

An entire PPP interface can be removed with the command:

```
DESTROY PPP=interface
```

Additional physical interfaces can be added to the PPP interface to form a multilink bundle, using the command:

```
ADD PPP=interface OVER=physical-interface
```

If an ISDN call is being added as the physical interface, multiple physical interfaces can be added using the NUMBER parameter. For example, the following command adds two identical ISDN calls (named "HeadOffice") as physical interfaces to PPP interface 0:

```
ADD PPP=0 OVER=ISDN-HeadOffice NUM=2
```

Members of a multilink bundle can be selectively deleted with the command:

```
DELETE PPP=interface OVER=physical-interface
```

An entire PPP interface can be temporarily disabled or enabled, or reset, with the commands:

```
DISABLE PPP=interface
ENABLE PPP=interface
RESET PPP=interface
```

One of the features of PPP is the negotiation of options for each protocol using the link. All options have a default value to which the option will be set if either end of the PPP link does not wish the option to be different from the default. The LCP will attempt to negotiate the Maximum Receive Unit (MRU), Authentication Protocol, Link Quality Reporting (LQR), Magic Number and Maximum Received Reconstructed Unit (MRRU) options. All other possible options are set to the default values specified in the relevant RFC.

The IP NCP will attempt to negotiate Van Jacobson's TCP/IP header compression if this has been turned on with the command:

```
ADD IP INTERFACE... VJC=ON
```

For more information on turning on Van Jacobson's TCP/IP header compression, see *Chapter 6, Internet Protocol (IP)*.



Van Jacobson's TCP/IP header compression should not be enabled on a multilink PPP interface.

The IP NCP will also negotiate the IP Address option. This option is used to inform each end of the link what the IP address of the other end of the link is by passing the address to the peer inside the option.

If the PPP interface has an IP address of 0.0.0.0 defined it may request an IP address from the peer by passing an IP address of 0.0.0.0 to the peer. If the peer has an IP address to allocate it will pass this IP address to the requesting router in a IPCP Configure Nak packet. To configure the router to request an IP address using the IP address option, use the commands:

```
SET PPP=ppp-interface IPREQUEST=ON
SET IP INT=ppp-interface IPADDRESS=0.0.0.0
ENABLE IP REMOTEASSIGN
RESET IP
```

The IP NCP also provides a number of options for requesting name server addresses from the peer. These name server addresses consist of primary and secondary DNS and WINS (Windows Internet Name Service) server addresses.

The router will only request the primary DNS address from a peer, but will supply the peer with primary and secondary DNS and WINS server addresses if a request is made. The values to be supplied to the peer are set using the command:

```
SET PPP [DNSPRIMARY=ipadd] [DNSSECONDARY=ipadd]
      [WINSPRIMARY=ipadd] [WINSSECONDARY=ipadd]
```

The router supports both the CHAP and PAP authentication protocols through the AUTHENTICATION parameter of the ADD PPP command, the CREATE PPP command and the SET PPP command:

```
CREATE PPP=ppp-interface OVER=physical-interface
      AUTHENTICATION={NONE | PAP | CHAP | EITHER}
ADD PPP=ppp-interface OVER=physical-interface
      AUTHENTICATION={NONE | PAP | CHAP | EITHER}
SET PPP=ppp-interface OVER=physical-interface
      AUTHENTICATION={NONE | PAP | CHAP | EITHER}
```

The router supports the link quality management options for PPP through the LQR parameter of the ADD PPP command, the CREATE PPP command and the SET PPP command:

```
ADD PPP=0 OVER=physical-interface LQR=ON
```

A PPP interface can be configured for dial-on-demand operation by specifying the IDLE parameter in the CREATE PPP command or the SET PPP command:

```
CREATE PPP=0 OVER=ISDN-HeadOffice NUM=2 IDLE=ON
```

A PPP interface can be configured for bandwidth on demand by specifying the TYPE parameter in the ADD PPP and the CREATE PPP commands, and the UPRATE, UPTIME, DOWNRATE and DOWNTIME parameters in the CREATE PPP and SET PPP commands:

```
CREATE PPP=0 OVER=ISDN-HeadOffice NUM=2 IDLE=ON TYPE=DEMAND
      UPRATE=60 UPTIME=30 DOWNRATE=20 DOWNTIME=30
```

A PPP interface can be configured to provide AODI (*Always On/Dynamic ISDN*) by specifying a MIOX circuit as the primary link and an ISDN call as the demand link in the ADD PPP and CREATE PPP commands:

```
CREATE PPP=0 OVER=MIOX3-AODI IDLE=40000000
ADD PPP=0 OVER=ISDN-AODI TYPE=DEMAND NUM=2
```

See “*Always On/Dynamic ISDN (AODI)*” on page 4-28 of *Chapter 4, Integrated Services Digital Network (ISDN)* for more information about configuring AODI.

The router uses a number of counters and timers to control the LCP and NCPs. The timers control the retransmission of *Configure-Request* and *Terminate-Request* control protocol packets. If the correct acknowledgement is not seen in the timeout period, another packet is transmitted. Counters control the number of times the packets can be sent. The *Configure* counter records retransmissions of *Configure-Requests*. If this counter exceeds the value set for it, the LCP will reset the interface and start again. The *Terminate* counter records retransmissions of *Terminate-Requests*. If this counter exceeds the value set for it, the link is assumed to be DOWN. The *Failure* counter is used to control the number of attempts to reach an agreeable set of values for options being negotiated by an NCP. This counter is not used in the router.

The values for the timers and counters can be set with the ADD, CREATE or SET commands.

Interface parameters can be modified after the interface has been created, with the command:

```
SET PPP=interface parameter=value...
```

The command:

```
SHOW PPP[=interface] [CONFIGURATION|COUNT|IDLETIMER|
MULTILINK]
```

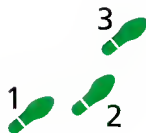
displays information about a PPP interface. If CONFIGURATION is specified, the settings of configuration parameters such as LQR and restart timers are displayed. If COUNT is specified, counters from the interface MIB and counters for the users of the interface are displayed. If IDLETIMER is specified, the configured and current values of the idle timer are displayed. If MULTILINK is specified, information about the multilink bundle associated with the interface is displayed. The display includes the number of links in the bundle, the number of packets fragmented, the number of packets or fragments in the multilink receive queue, and information about the sequence numbers on the multilink bundle. If no optional parameters are specified, a summary of the configured PPP interfaces, the physical interfaces used and the Network Control Protocols (NCPs) in use is displayed.

Configuration Examples

The following examples illustrate some of the options available for configuring PPP interfaces to provide a range of network services.

Configuring a PPP link

In this example, a Point-to-Point Protocol (PPP) link will be set up between two routers. The function of PPP is to maintain a channel between the routers, over which data can be exchanged. To exchange data, the relevant routing module(s) must be assigned to use a PPP link.



To configure a PPP link:

1. Create the PPP interface.

On Router A, create a PPP interface numbered 0 over ISDN call "remote":

```
CREATE PPP=0 OVER=ISDN-remote
```

On Router B, create a PPP interface numbered 1 over ISDN call "remote":

```
CREATE PPP=1 OVER=ISDN-remote
```

The PPP interface is enabled by default when it is created. The ISDN call must have been created previously. See *Chapter 4, Integrated Services Digital Network (ISDN)* for more information about defining ISDN calls.

Additional physical interfaces can be added to the PPP interface to form a multilink bundle, using the ADD PPP command on page 3-28. To add the ISDN call "demand" as a physical interface to the PPP interface created above, on Router A use the command:

```
ADD PPP=0 OVER=ISDN-demand
```

On Router B, use the command:

```
ADD PPP=1 OVER=ISDN-demand
```


The PPP interface may be configured for dial on demand operation by adding the IDLE=ON option to the CREATE commands, or the option may be set at a later date with the SET command:

```
SET PPP=0 IDLE=ON
```

2. Enable routing modules to use the interface.

Once a PPP interface has been defined and configured, routing modules can be configured to use the interface. The procedures for achieving this are described in the chapter for the particular routing module.

In general, commands that contain the parameter INTERFACE= can refer to a PPP interface by name. The form of the name is "pppn", where *n* is the interface number for the PPP module. For example:

```
ADD IP INTERFACE=PPPN...
```

As an example, the IP routing module is to use the PPP interface just configured. The RIP routing protocol is to be used, so the PPP link has to be assigned its own an IP subnet. The subnet assigned to the PPP link is 172.16.254.0, with 255.255.255.0 as the subnet mask. The local (Router A) end of the link will have address 172.16.254.1, and the remote (Router B) end will have address 172.16.254.2. RIP is to be enabled to the remote end of the link. Router A already has an IP interface for the Ethernet interface, with an IP address of 172.16.9.59. The commands for Router A are:

```
ENABLE IP
ADD IP INT=PPP0 IP=172.16.254.1 MASK=255.255.255.0
ADD IP RIP INT=PPP0
```

3. Test that the link is active.

The PPP interface can be checked with the command:

```
SH PPP
```

which produces a display like Figure 3-3 on page 3-21. For each control protocol (listed in the *CP* field), the corresponding *State* field should be set to 'OPENED'.

Figure 3-3: Example output from the SHOW PPP command for a PPP link.

Name	Enabled	ifIndex	Over	CP	State
-----	-----	-----	-----	-----	-----
ppp0	YES	04		IPCP	OPENED
			isdn-remote	LCP	OPENED
			isdn-demand	LCP	OPENED
-----	-----	-----	-----	-----	-----

If the LCP has a state that is not 'OPENED' check the configuration of the physical interfaces used by the PPP interface. Check that the local NTUs are the correct type. Contact your dealer for assistance.

Check the NTUs are correctly installed. Perform a remote loop back from each end alternately. Contact the Telecom supplier or your dealer if this fails.

For a PPP interface that is using an ISDN call as the physical interface, check that the calls have been properly defined and are active on the routers at each end of the link.

If the routing protocol is not in the state 'OPENED' check the configuration of the routing module. As a first step, the IP configuration can be checked with the command:

```
SHOW IP INTERFACE
```

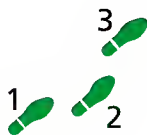
which produces a display like Figure 3-4 on page 3-22.

Figure 3-4: Example output from the **SHOW IP INTERFACE** command for a PPP link configured for use by the IP routing module.

Interface Pri. Filt	Type Pol.Filt	IP Address Network Mask	Bc Fr PArp MTU VJC	Filt GRE	RIP Met. DBcast Mul.
-----	-----	-----	-----	-----	-----
LOCAL	-	Not Set	-	-	---
---	---	---	---	---	---
eth0	Static	202.36.163.36	1 On	---	01
---	---	255.255.255.0	1500 -	---	None
ppp0	Static	192.168.1.1	1 -	---	01
---	---	255.255.255.0	1500 Off	---	None
ppp1	Static	192.168.2.1	1 -	---	01
---	---	255.255.255.0	1500 Off	---	None
-----	-----	-----	-----	-----	-----

Multilink Aggregation

Traffic can be sent over multiple physical interfaces using PPP multilink. A PPP interface is created and configured to use more than one physical interface (e.g. ISDN B channels). This example expands on "A Basic ISDN Setup" on page 4-33 of *Chapter 4, Integrated Services Digital Network (ISDN)*, by aggregating traffic on two ISDN B channels between router HO1 and RG1.



To configure channel aggregation on a PPP interface:

1. Set up the ISDN call.

Create an ISDN call between routers HO1 and RG1 as in "A Basic ISDN Setup" on page 4-33 of *Chapter 4, Integrated Services Digital Network (ISDN)*.

2. Create a PPP interface to use the ISDN call.

Create a PPP interface to use the ISDN call Region1 twice (i.e. activate two calls using the same call definition). On the Head Office router, create PPP0 to use ISDN call Region1:

```
CREATE PPP=0 OVER=ISDN-Region1 NUM=2 IDLE=ON
```

On the Region 1 router, create PPP0 to use the ISDN call HeadOffice twice:

```
CREATE PPP=0 OVER=ISDN-Region1 NUM=2 IDLE=ON
```

3. Configure routing modules to use the PPP interface.

Configure one or more routing modules to use the PPP interface. See "A Basic ISDN Setup" on page 4-33 of *Chapter 4, Integrated Services Digital Network (ISDN)*.

4. Test the configuration.

The PPP configuration can be checked using the command:

```
SHOW PPP
```

The expected output is shown in Figure 3-5 on page 3-23. All control protocols should have their State set to 'OPENED'. If either PPP LCP is not

in the 'OPENED' state, check that the ISDN calls are active on both routers. If any of the routing control protocols (in this case IPCP) is not in the 'OPENED' state check the configuration of the routing module on both routers.

Figure 3-5: Example output from a SHOW PPP command for a PPP interface aggregated over two ISDN B channels.

Name	Enabled	ifIndex	Over	CP	State

ppp0	YES	04		IPCP	OPENED
			isdn-Region1	LCP	OPENED
			isdn-Region1	LCP	OPENED

The ISDN calls can be checked using the command:

```
SHOW ISDN CALL
```

The expected output is shown in Figure 3-6 on page 3-23. There should be two active calls with the State field set to 'ON'. If not, the calls can be attempted again either by deactivating and then reactivating them, or by resetting the interface. For the HO1 router the commands are:

```
DEACTIVATE ISDN CALL=Region1
ACTIVATE ISDN CALL=Region1
ACTIVATE ISDN CALL=Region1
```

or:

```
RESET PPP=0
```



*The DEACTIVATE ISDN CALL command deactivates **all** calls with the specified name. In this example, PPP has been configured to make two Region1 calls. The DEACTIVATE ISDN CALL command will deactivate (hang up) both calls. The ACTIVATE ISDN CALL command makes a single call based on the specified call definition. To reactivate both calls for this example, the ACTIVATE ISDN CALL command must be used twice.*

Figure 3-6: Example output from the SHOW ISDN CALL command for a PPP interface aggregated over two ISDN B channels.

ISDN call details				
Name	Number	Remote call	State	Precedence

Region1	043332345	-	IN & OUT	OUT

ISDN active calls				
Index	Name	User	State	Prec

0	Region1	03-00	ON	Yes
1	Region1	03-36	ON	Yes

Dial on Demand Links

A PPP interface can be configured so that it only brings the link up when there is traffic to send. This feature is only useful on switched interfaces (e.g. ISDN) because for other types the physical layer is available all the time. This feature is sometimes called dial on demand. The link is disconnected when there has been no traffic for a specified period of time. This feature is disabled by default. The follow examples assume PPP interface 0 has been configured as in "A Basic ISDN Setup" on page 4-33 of *Chapter 4, Integrated Services Digital Network (ISDN)*.

To enable dial-on-demand and use the default disconnect timer (60 seconds), use the command:

```
SET PPP=0 IDLE=ON
```

To enable dial-on-demand with the disconnect timer set to 20 seconds, use the command:

```
SET PPP=0 IDLE=20
```

To disable dial-on-demand, use the command:

```
SET PPP=0 IDLE=OFF
```

To check the configuration, use the command:

```
SHOW PPP=0 CONF
```

Link Quality Monitoring

Link quality monitoring is used to measure the quality of a link. The protocol used is an option negotiated when the link is brought up. There is only one protocol for this, Link Quality Report. Packet and octets loss count, and link failure can be determined using LQR. The negotiation process determines how often a router should receive an LQR packet on a PPP interface. If a router does not receive two consecutive LQR packets within the specified time frame it will reset the link. When using an ISDN call with the PPP interface this will disconnect the call if it is connected and try to reconnect it.

The LQR counters can be displayed with the command:

```
SHOW PPP=0 COUNT
```

and the network manager can decide whether the level of packet and octet loss is good or bad. In a multilink configuration, LQR can be configured differently on each physical interface in the multilink bundle. The following examples assume PPP interface 0 has been configured as in "A Basic ISDN Setup" on page 4-33 of *Chapter 4, Integrated Services Digital Network (ISDN)*.

To enable LQR with the default timer (60 seconds), use the command:

```
SET PPP=0 OVER=ISDN-HeadOffice LQR=ON
```

To enable LQR with the timer set to 20 seconds, use the command:

```
SET PPP=0 OVER=ISDN-HeadOffice LQR=20
```

To disable LQR, use the command:

```
SET PPP=0 OVER=ISDN-HeadOffice LQR=OFF
```

To check the configuration, use the command:

```
SHOW PPP=0 CONF
```

Compression

PPP interfaces can be configured to compression over wide area links. See *Chapter 8, Compression Services* for more information. Compression must be configured on per-interface basis, on the routers at both ends of the PPP link.

To enable compression, use the commands:

```
CREATE PPP=0 OVER=physical-interface COMP=ON
```

or:

```
SET PPP=0 COMP=ON
RESET PPP=0
```

To disable compression, use the commands:

```
SET PPP=0 COMP=OFF
RESET PPP=0
```

To check any of these configurations, use the command:

```
SHOW PPP CONFIG
```

Bandwidth on Demand

A PPP interface can be configured to use up to two B channels on a Basic Rate ISDN interface, or up to 30 B channels on a Primary Rate ISDN interface, to provide bandwidth on demand. PPP activates channels when the bandwidth used exceeds an upper threshold and deactivates channels when the bandwidth used drops below a lower threshold. To configure bandwidth on demand the ISDN channels are assigned a TYPE of DEMAND when added to the PPP interface. Assigning one channel a type of PRIMARY and other channels a TYPE of DEMAND will ensure that there is always one channel available. If all channels are assigned a TYPE of DEMAND then there will be no channels active when there is no traffic; some traffic will cause one channel to be activated and continuous traffic will cause other channels to be activated. If there is one channel remaining opened then the IDLE timer is used to determine when this should be closed. In this case the IDLE timer should not be set to OFF.

This example illustrates how to configure bandwidth on demand between two routers (Figure 3-7 on page 3-25, Table 3-3 on page 3-26).

Figure 3-7: Example configuration for bandwidth on demand.

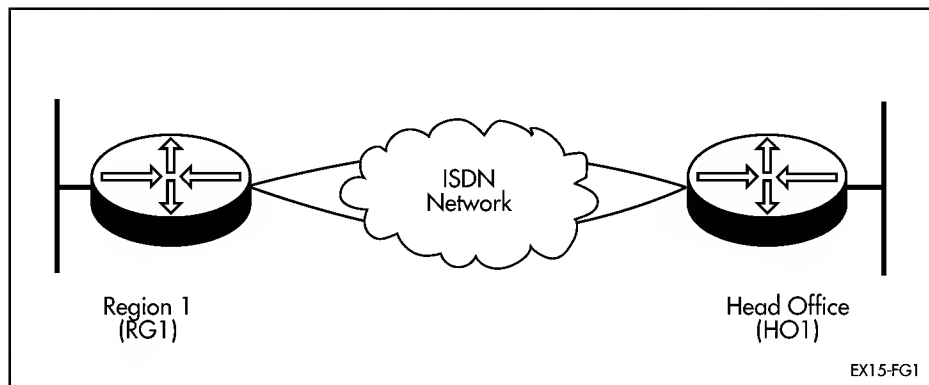
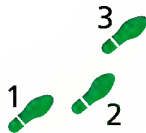


Table 3-3: Example configuration parameters for bandwidth on demand.

Site	Region 1	Head Office
Router Name	RG1	HO1
ISDN Number	1234567	9876543
IP Address for PPP0	192.168.35.114	192.168.35.113
IP Address for Eth0	192.168.35.110	192.168.35.45
Subnet Mask	255.255.255.240	255.255.255.240

**To configure PPP for bandwidth on demand:****1. Create the ISDN calls.**

An ISDN call must be defined on each router so that either router may initiate a call to transfer data. For a more detailed example of creating ISDN calls see “A Basic ISDN Setup” on page 4-33 of *Chapter 4, Integrated Services Digital Network (ISDN)*.

Set the ISDN call profile appropriate for the ISDN service provider. The default profile is the ETSI specification for European Union (EU) countries (ETB for Basic Rate interfaces or ETP for Primary Rate interfaces). To use the Australian Telecom profile, for example, on Basic Rate interface BRI 0 for router HO1 and RG1, use the following command on each router:

```
SET Q931=BRI0 PROFILE=AUS
```

On the Head Office router, create calls to the Region 1 router:

```
ADD ISDN CALL=Region1 PREC=IN OUTSUB=LOCAL SEARCHSUB=LOCAL
NUMBER=1234567
ADD ISDN CALL=Demand PREC=IN OUTSUB=LOCAL SEARCHSUB=LOCAL
NUMBER=1234567
```

On the Region 1 router create calls to the Head Office router:

```
ADD ISDN CALL=Region1 PREC=OUT OUTSUB=LOCAL
SEARCHSUB=LOCAL NUMBER=9876543
ADD ISDN CALL=Demand PREC=OUT OUTSUB=LOCAL SEARCHSUB=LOCAL
NUMBER=9876543
```

2. Create a PPP interface to use the ISDN calls.

Create the PPP interface with one primary channel and one demand channel. The primary channel is created with the IDLE parameter ON (defaults to 60 seconds). The demand channel is added with the TYPE parameter set to DEMAND. On the Head Office router create a PPP interface:

```
CREATE PPP=0 OVER=ISDN-Region1 IDLE=ON
ADD PPP=0 OVER=ISDN-Demand TYPE=DEMAND
```

On the Region 1 router create a PPP interface:

```
CREATE PPP=0 OVER=ISDN-Region1 IDLE=ON
ADD PPP=0 OVER=ISDN-Demand TYPE=DEMAND
```

3. Configure IP.

Configure IP to use the PPP interfaces. Static routes must be defined with on-demand links because a routing protocol would keep a link up continuously. Configure IP at the Head Office router:

```
ENABLE IP
ADD IP INT=ppp0 IP=192.168.35.113 MASK=255.255.255.240
ADD IP ROUTE=192.168.35.96 INT=ppp0 NEXT=192.168.35.114
```

MET=2

Configure IP at the Region 1 router:

```
ENABLE IP
ADD IP INT=ppp0 IP=192.168.35.114 MASK=255.255.255.240
ADD IP ROUTE=192.168.35.0 INT=ppp0 NEXT=192.168.35.113
      MET=2 MASK=255.255.255.0
ADD IP ROUTE=0.0.0.0 INT=ppp0 NEXT=192.168.35.113 MET=3
```

For a more detailed example of configuring IP see “*Configuration Examples*” on page 6-33 of *Chapter 6, Internet Protocol (IP)*.

Command Reference

This section describes the commands available on the router to configure and manage the Point-to-Point Protocol on the router. The Point-to-Point Protocol (PPP) can be used on ISDN calls (see *Chapter 4, Integrated Services Digital Network (ISDN)*), MIOX circuits (see *Chapter 5, X.25*), TDM groups (see *Chapter 11, Time Division Multiplexing (TDM)*) and Ethernet. PPP can be used to carry IP and compressed data.

See “*Conventions*” on page xxxv of *Preface* in the front of this manual for details of the conventions used to describe command syntax. See *Appendix A, Messages* for a complete list of messages and their meanings.

ACTIVATE PPP

Syntax `ACTIVATE PPP=ppp-interface RXPKT=hexstring`

where:

- *ppp-interface* is the PPP interface number.
- *hexstring* is a string of hexadecimal characters.

Description This command creates and sends a PPP packet to the specified PPP interface as if the packet had been received from the lower layer interface, and is intended for diagnostic and testing purposes.

The RXPKT parameter specifies the PPP packet to create and send, as a string of hexadecimal characters. For detailed information about PPP packet formats, see RFC 1661, “*The Point-to-Point Protocol (PPP)*”.



This command is intended for debugging purposes only, and should not be used during normal operation.

Examples To create and send an LCP packet requesting CHAP authentication to PPP interface 1, use the command:

```
ACTIVATE PPP=1 RXPKT=ff03c023012100090305c22305
```

ADD PPP

Syntax ADD PPP=*ppp-interface* OVER=*physical-interface*
 [AUTHENTICATION={CHAP|EITHER|PAP|NONE}] [AUTHMODE={IN|OUT|INOUT}] [CBDELAY=1..100] [CBMODE={ACCEPT|OFF|REQUEST}] [CBNUMBER=*e164number*]
 [CBOperation={E164NUMBER|USERAUTH}]
 [COMPALGORITHM=STACLS] [COMPRESSION={LINK|OFF}]
 [CONFIGURE={*value*|CONTINUOUS}] [LQR={ON|OFF|*time*}]
 [MAGIC={ON|OFF}] [NUMBER=*number*] [RESTART=*time*]
 [STACHECK={LCB|SEQUENCE}] [TERMINATE={*value*|CONTINUOUS}] [TYPE={DEMAND|PRIMARY|SECONDARY}]

where:

- *ppp-interface* is the PPP interface number.
- *physical-interface* is ISDN-callname, MIOXn-circuitname, TDM-groupname or ETHn-servicename.
- *e164number* is the phone number to dial when performing callback. It may contain digits (0–9) and should be a valid phone number as described in CCITT standard E.164.
- *value* is a retry threshold.
- *time* is a timer value in seconds.
- *number* is the number of PPP interfaces to add.

Description This command adds an ISDN call, a MIOX circuit, TDM group or a PPP over Ethernet service to the PPP interface to use as a physical layer. The OVER parameter specifies the physical interface over which the PPP interface will run.

The AUTHENTICATION parameter specifies the authentication protocol to be used on the physical interface or channel. If CHAP is specified, the Challenge-Handshake Authentication Protocol (CHAP) is used. If PAP is specified, the Password Authentication Protocol (PAP) is used. If EITHER is specified, the router uses the option negotiation process to negotiate the authentication protocol to be used with the device at the remote end of the link, specifying CHAP as the first choice. If NONE is specified, no authentication protocol is used. The default is NONE.

The AUTHMODE parameter specifies how authentication requests to peers are affected by the direction of the ISDN call. The AUTHMODE parameter is only valid when the AUTHENTICATION parameter is set to a value other than NONE and the physical interface is an ISDN call. If IN is specified, authentication will only be requested for incoming calls from peers. If OUT is specified, authentication will only be requested for outgoing calls to peers. If INOUT is specified, authentication will always be requested regardless of the direction of the call. The default is INOUT.

The CBDELAY parameter specifies the delay, in tenths of a second, between bringing down a call for callback and actually making the call back to the peer. This parameter is used to handle the different timing requirements of various ISDN switches and is only valid for PPP links over ISDN calls and when the callback mode is REQUEST. The default is 1.

The CBMODE parameter specifies whether a callback request will be made or accepted during the LCP negotiation. If REQUEST is specified a request will be

made for callback. If ACCEPT is specified requests for callback received from the peer will be accepted and processed, however AUTHENTICATION must be set to PAP or CHAP. If OFF is specified no callback requests will be made and callback requests will not be accepted. The default is OFF.

The CBNUMBER parameter specifies the number to include when requesting a callback with the CBOperation parameter set to E164NUMBER. The number specified should be a phone number as specified in the E.164 standard.

The CBOperation parameter specifies the callback operation to be included in the callback request to specify to the peer how to determine the callback number. If USERAUTH is specified the peer will use the username and password supplied during authentication to look up the callback number. If E164NUMBER is specified the callback number specified by the CBNUMBER parameter is included in the callback request. The default is USERAUTH.

The COMPALGORITHM parameter specifies the compression algorithm to use when compressing and decompressing PPP packets. If STACLZS is specified the Stac LZS algorithm will be used as specified in RFC 1974. The default is STACLZS.

The COMPRESSION parameter enables compression for the physical interface being added. The default is OFF. The LINK option should only be used when compression is required on the interface being added and not on others. If compression is required on all physical interfaces of a PPP interface, compression should be enabled by setting the COMPRESSION parameter to ON in the CREATE PPP command on page 3-31 or the SET PPP command on page 3-49.

The CONFIGURE parameter sets the number of configure requests sent before some action is taken. For the LCP the action is to reset the hardware and start again. For all other protocols the action is to give up. The default is CONTINUOUS, which means that requests will be sent continuously.

The LQR parameter sets the LQR timer. If ON is specified, LQR is enabled with a default timer of 60 seconds. If a time is specified, LQR is enabled with the timer set to the specified time. If OFF is specified, LQR is disabled. The default is ON.

The MAGIC parameter enables or disables negotiation of the magic number option. The default is ON. The magic number is used to determine if a interface is looped back. The interface will not reach the OPENED state if there is a loopback.

The NUMBER parameter specifies the number of physical interfaces to be added. This parameter is only valid when the OVER parameter specifies an ISDN call as the physical interface. The default is 1.

The RESTART parameter specifies the time between successive retransmissions of unacknowledged configure requests or terminate requests. The default is 3 seconds.

The STACCHECK parameter specifies the check mode to used for the Stac LZS compression algorithm. If SEQUENCE is specified an incrementing sequence number is used to determine whether a packet has been lost and therefore whether the compression history needs to be reset. If LCB is specified an LCB value is used to determine if an error has occurred in a packet. The default is SEQUENCE.

The TERMINATE parameter sets the number of terminate requests sent when trying to close a link before it is assumed the link is down. The default is 2. The CONTINUOUS option specifies that requests will be sent continuously.

The TYPE parameter specifies the role of the physical interface for bandwidth on demand and leased line backup. The default is PRIMARY. If PRIMARY is specified, the link will be kept open all the time (IDLE=OFF) or opened whenever there is traffic (IDLE=ON). If SECONDARY is specified, the link will be opened only when the associated primary link fails. If DEMAND is specified, the link will be opened only when additional bandwidth is required.

To configure bandwidth on demand the ISDN channels are given a TYPE of DEMAND when they are added to the PPP interface. Adding one channel with a type of PRIMARY and other channels with a TYPE of DEMAND will ensure that there is always one channel available. If all channels are assigned a TYPE of DEMAND then there will be no channels open when there is no traffic, some traffic will cause one channel to be opened and continuous traffic will cause other channels to be opened. If there is one channel remaining opened then the IDLE timer is used to determine when this channel should be closed. For bandwidth on demand the IDLE timer should not be set to OFF.

Examples To add ISDN call “demand” as an additional physical interface to PPP interface 1, and enable STAC LZS compression on the link with a check mode of LCB, use the command:

```
ADD PPP=1 OVER=ISDN-demand COMP=LINK STACHECK=LCB
```

See Also CREATE PPP
 DELETE PPP
 DESTROY PPP
 SET PPP
 SHOW PPP

CREATE PPP

Syntax `CREATE PPP=ppp-interface OVER=physical-interface`
`[AUTHENTICATION={CHAP|EITHER|PAP|NONE}] [AUTHMODE={IN|`
`OUT|INOUT}] [BAP={ON|OFF}] [BAPMODE={CALL|CALLBACK}]`
`[CBDELAY=1..100] [CBMODE={ACCEPT|OFF|REQUEST}]`
`[CBNUMBER=e164number] [CBOperation={E164NUMBER|`
`USERAUTH}] [COMPALGORITHM=STACLS] [COMPRESSION={ON|`
`OFF|LINK}] [CONFIGURE={value|CONTINUOUS}]`
`[DEBUGMAXBYTES=16..256] [DESCRIPTION=description]`
`[DOWNRATE=0..100] [DOWNTIME=time] [ECHO={ON|OFF|`
`period}] [FRAGMENT={ON|OFF}] [FRAGOVERHEAD=0.100]`
`[IDLE={ON|OFF|time}] [INDATALIMIT={NONE|1..65535}]`
`[IPPOOL={pool-name|NONE}] [IPREQUEST={ON|OFF}]`
`[LQR={ON|OFF|time}] [MAGIC={ON|OFF}]`
`[NULLFRAGTIMER=time] [NUMBER=number]`
`[ONLINELIMIT={NONE|1..65535}] [OUTDATALIMIT={NONE|`
`1..65535}] [PASSWORD=password] [RESTART=time]`
`[STACCHECK={LCB|SEQUENCE}] [TERMINATE={value|`
`CONTINUOUS}] [TOTALDATALIMIT={NONE|1..65535}]`
`[TYPE={DEMAND|PRIMARY|SECONDARY}] [UPRATE=0..100]`
`[UPTIME=time] [USERNAME=username]`

where:

- *ppp-interface* is the PPP interface number.
- *physical-interface* is ISDN-callname, MIOXn-circuitname, TDM-groupname or ETHn-servicename.
- *e164number* is the phone number to dial when performing callback. It may contain digits (0–9) and should be a valid phone number as described in CCITT standard E.164.
- *value* is a retry threshold.
- *description* is a character string, 1 to 70 characters in length. Valid characters are any printable character.
- *time* is a timer value in seconds.
- *period* is a decimal number in the range 1 to 4294967295.
- *pool-name* is a character string, 1 to 15 characters in length. Valid characters are any printable characters. If *pool-name* contains spaces, it must be enclosed in double quotes.
- *number* is the number of PPP interfaces to create.
- *password* is the password to use for authentication, 1 to 32 characters in length. It may contain any printable character, and is case sensitive.
- *username* is the username to use for authentication, 1 to 32 characters in length. It may contain any printable character, and is case sensitive.

Description This command creates the specified PPP interface running over an ISDN call, a MIOX circuit, a TDM group (referred to as a physical layer) or a PPP over Ethernet service.

The OVER parameter specifies the physical interface over which the PPP interface will run. Additional physical interfaces can be added to the PPP interface using the ADD PPP command on page 3-28.

The AUTHENTICATION parameter specifies the authentication protocol to be used on the physical interface or channel. If CHAP is specified, the Challenge-Handshake Authentication Protocol (CHAP) is used. If PAP is specified, the Password Authentication Protocol (PAP) is used. If EITHER is specified, the router uses the option negotiation process to negotiate the authentication protocol to be used with the device at the remote end of the link, specifying CHAP as the first choice. If NONE is specified, no authentication protocol is used. The default is NONE.

The AUTHMODE parameter specifies how authentication requests to peers are affected by the direction of the ISDN call. The AUTHMODE parameter is only valid when the AUTHENTICATION parameter is set to a value other than NONE and the physical interface is an ISDN call. If IN is specified, authentication will only be requested for incoming calls from peers. If OUT is specified, authentication will only be requested for outgoing calls to peers. If INOUT is specified, authentication will always be requested regardless of the direction of the call. The default is INOUT.

The BAP parameter specifies whether or not the Bandwidth Allocation Protocol will be used for negotiating the activation of demand PPP links. The default is ON.

The BAPMODE parameter specifies which peer originates another link to add to the multilink bundle. For CALLBACK mode, the number to call must be configured on the call at the lower layer (ISDN). The default is CALL.

The CBDELAY parameter specifies the delay, in tenths of a second, between bringing down a call for callback and actually making the call back to the peer. This parameter is used to handle the different timing requirements of various ISDN switches and is only valid for PPP links over ISDN calls and when the callback mode is REQUEST. The default is 1.

The CBMODE parameter specifies whether a callback request will be made or accepted during the LCP negotiation. If REQUEST is specified a request will be made for callback. If ACCEPT is specified requests for callback received from the peer will be accepted and processed, however AUTHENTICATION must be set to PAP or CHAP. If OFF is specified no callback requests will be made and callback requests will not be accepted. The default is OFF.

The CBNUMBER parameter specifies the number to include when requesting a callback with the CBOperation parameter set to E164NUMBER. The number specified should be a phone number as specified in the E.164 standard.

The CBOperation parameter specifies the callback operation to be included in the callback request to specify to the peer how to determine the callback number. If USERAUTH is specified the peer will use the username and password supplied during authentication to look up the callback number. If E164NUMBER is specified the callback number specified by the CBNUMBER parameter is included in the callback request. The default is USERAUTH.

The COMPALGORITHM parameter specifies the compression algorithm to use when compressing and decompressing PPP packets. If STACLZS is specified the Stac LZS algorithm will be used as specified in RFC 1974. The default is STACLZS.

The COMPRESSION parameter enables or disables the use of compression for the interface. When used with multilink, setting COMPRESSION to ON will compress the packets before they are sent to the individual links. Setting COMPRESSION to LINK will enable compression for the link specified by the

OVER parameter. The default is OFF. The LINK option should only be used when compression is required on some physical interfaces and not on others. If compression is required on all physical interfaces of a PPP interface, the COMPRESSION parameter should be set to ON.

The CONFIGURE parameter sets the number of configure requests sent before some action is taken. For the LCP the action is to reset the hardware and start again. For all other protocols the action is to give up. The default is CONTINUOUS, which means that requests will be sent continuously.

The DEBUGMAXBYTES parameter specifies the maximum number of bytes that are displayed for each packet when the PACKET debug option is enabled. The default is 32.

The DESCRIPTION parameter specifies a user-defined description for the interface, to make it easier to distinguish between a number of PPP interfaces.

The DOWNTIME parameter specifies the time, in seconds, that the PPP interface must have a total utilisation (as a percentage) below the threshold specified by the DOWNRATE parameter, before a channel is closed. The default is 60 for DOWNTIME and 20 for DOWNRATE. The UPRATE, UPTIME, DOWNRATE and DOWNTIME parameters are used in conjunction with the TYPE parameter to configure bandwidth on demand.

The ECHO parameter specifies whether or not LCP *Echo Request* and *Echo Reply* messages are used to determine link quality. If three consecutive *Echo Request* messages are transmitted without receiving an *Echo Reply* response, the link is deemed to be down. The ECHO and LQR parameters are mutually exclusive. If ECHO is enabled, LQR will be disabled. If LQR is enabled, ECHO will be set to OFF. If OFF is specified, *Echo Request* messages will not be transmitted. If ON is specified, *Echo Request* messages will be transmitted every 60 seconds. If a period in seconds is specified, *Echo Request* messages are transmitted at the specified interval.

The FRAGMENT parameter applies only to a multilink bundle interface, and determines whether packets are fragmented or not. The default is OFF. Fragmentation must be disabled if compression is required.

The FRAGOVERHEAD parameter specifies the maximum allowable overhead, as a percentage, for fragmenting packets using the variable fragmentation scheme for multilink PPP. If this limit will be exceeded for any packet the packet is fragmented using the fixed fragmentation scheme. The default is 5. The variable fragmentation scheme spreads the packet over all the links in the multilink bundle by splitting the packet into variable sized fragments to match the speed of individual links. Larger fragments are transmitted over faster links, thereby providing an inherent load balancing scheme. The fixed fragmentation scheme spreads the packet over all the links in the multilink bundle by splitting the packet into equal fixed sized fragments. If the number of links is large and the packet is relatively small a fragment is not transmitted over every link.

The IDLE parameter controls the dial-on-demand feature. If ON is specified, dial-on-demand is enabled with a default timer of 60 seconds. If a time is specified, dial-on-demand is enabled with the timer set to the specified time. If OFF is specified, dial-on-demand is disabled. When the dial-on-demand feature is activated, PPP brings up the link when there is traffic to be sent, and takes down the link when there has been no traffic for the specified timer period. The effect on a PPP interface using an ISDN call will be to connect the call when traffic is to be sent and disconnect the call when no traffic has been

sent or received for the specified timer period. For other physical interfaces, this parameter has no effect, as the links are always connected. The default is OFF.

The INDATALIMIT parameter specifies the input data threshold, in megabytes, for the PPP interface. When the interface's cumulative input data counter exceeds this limit, the link is closed and any further attempts to open the link will fail. If the input data counter for the interface is cleared using the RESET PPP command on page 3-48, the link can be reopened. The default is NONE, which sets no threshold.

The IPOOL parameter specifies the IP pool to use to allocate IP addresses for the remote end of the PPP connection. If NONE is specified, IP addresses are not allocated from an IP pool. The default is NONE. See "IP Address Pools" on page 6-32 of *Chapter 6, Internet Protocol (IP)* for more information about creating IP address pools.

The IPREQUEST parameter specifies whether or not a request will be made for an IP address to be allocated by the peer during the IPCP negotiation. If ON is specified a request will be made. If OFF is specified a request will not be specified. The default is OFF.

The LQR parameter sets the LQR timer. If ON is specified, LQR is enabled with a default timer of 60 seconds. If a time is specified, LQR is enabled with the timer set to the specified time. If OFF is specified, LQR is disabled. The default is ON.

The MAGIC parameter enables or disables negotiation of the magic number option. The default is ON. The magic number is used to determine if a interface is looped back. The interface will not reach the OPENED state if there is a loopback.

The NULLFRAGTIMER parameter specifies the maximum time, in seconds, a link in a multilink bundle may be idle before a NULL fragment is transmitted over the link. NULL fragments are used to keep the last sequence number transmitted over the link up to date. The default is 3.

The NUMBER parameter specifies the number of physical interfaces to be created. This parameter is only valid when the OVER parameter specifies an ISDN call as the physical interface. The default is 1.

The ONLINELIMIT parameter specifies the up-time threshold, in hours, for the PPP interface. When the interface's cumulative up-time counter exceeds this limit, the link is closed and any further attempts to open the link will fail. If the up-time counter for the interface is cleared using the RESET PPP command on page 3-48, the link can be reopened. The default is NONE, which sets no threshold.

The OUTDATALIMIT parameter specifies the output data threshold, in megabytes, for the PPP interface. When the interface's cumulative output data counter exceeds this limit, any further attempts to open the link will fail. If the output data counter for the interface is cleared using the RESET PPP command on page 3-48, the link can be reopened. The default is NONE, which sets no threshold.

The PASSWORD parameter specifies the password to use when the peer requests authentication using either CHAP or PAP. This is normally required for network lines between routers, for which an authentication protocol has been selected with the AUTHENTICATION parameter.

The **RESTART** parameter specifies the time between successive retransmissions of unacknowledged configure requests or terminate requests. The default is 3 seconds.

The **STACCHECK** parameter specifies the check mode to used for the Stac LZS compression algorithm. If **SEQUENCE** is specified an incrementing sequence number is used to determine whether a packet has been lost and therefore whether the compression history needs to be reset. If **LCB** is specified an LCB value is used to determine if an error has occurred in a packet. The default is **SEQUENCE**.

The **TERMINATE** parameter sets the number of terminate requests sent when trying to close a link before it is assumed the link is down. The default is 2. The **CONTINUOUS** option specifies that requests will be sent continuously.

The **TOTALDATALIMIT** parameter sets the total data throughput threshold, in megabytes, for the PPP interface. When the interface's cumulative total data counter exceeds this limit, the link is closed and any further attempts to open the link will fail. If the total data counter for the interface is cleared using the **RESET PPP** command on page 3-48, the link can be reopened. The default is **NONE** which corresponds to no threshold.

The **TYPE** parameter specifies the role of the physical interface for bandwidth on demand and leased line backup. The default is **PRIMARY**. If **PRIMARY** is specified, the link will be kept open all the time (**IDLE=OFF**) or opened whenever there is traffic (**IDLE=ON**). If **SECONDARY** is specified, the link will be opened only when the associated primary link fails. If **DEMAND** is specified, the link will be opened only when the additional bandwidth is required.

To configure bandwidth on demand the ISDN channels are given a **TYPE** of **DEMAND** when they are added to the PPP interface. Adding one channel with a type of **PRIMARY** and other channels with a **TYPE** of **DEMAND** will ensure that there is always one channel available. If all channels are assigned a **TYPE** of **DEMAND** then there will be no channels open when there is no traffic, some traffic will cause one channel to be opened and continuous traffic will cause other channels to be opened. If there is one channel remaining opened then the **IDLE** timer is used to determine when this channel should be closed. In this case it is recommended that the **IDLE** timer not be set to **OFF**.

The **UPTIME** parameter specifies the time, in seconds, that the PPP interface must have a total utilisation (as a percentage) above the threshold specified by the **UPRATE** parameter, before an additional channel is opened. The default is 30 for **UPTIME** and 80 for **UPRATE**. The **UPRATE**, **UPTIME**, **DOWNRATE** and **DOWNTIME** parameters are used in conjunction with the **TYPE** parameter to configure bandwidth on demand.

The **USERNAME** parameter specifies the username to be used when generating PAP authentication requests and when responding to CHAP authentication challenges. If the **USERNAME** is not set the router's system name will be used by default.



*For security reasons this command will only be accepted if the user has **SECURITY OFFICER** privilege.*

Examples To create PPP interface 0 with two on-demand channels over the ISDN call “ISDN-Region1”, use the command:

```
CREATE PPP=0 OVER=ISDN-Region1 IDLE=ON NUM=2 TYPE=DEMAND
```

See Also ADD PPP
DELETE PPP
DESTROY PPP
DISABLE PPP
ENABLE PPP
RESET PPP
SET PPP
SHOW PPP
SHOW PPP LIMITS

CREATE PPP TEMPLATE

Syntax CREATE PPP TEMPLATE=*template* [COPY=*template*]
[AUTHENTICATION={CHAP|EITHER|PAP|NONE}] [BAP={ON|OFF}]
[BAPMODE={CALL|CALLBACK}] [CBDELAY=1..100]
[CBMODE={ACCEPT|OFF|REQUEST}] [CBNUMBER=*e164number*]
[CBOperation={E164NUMBER|USERAUTH}]
[COMPALGORITHM=STACLS] [COMPRESSION={ON|OFF|LINK}]
[DEBUGMAXBYTES=16..256] [DESCRIPTION=*description*]
[ECHO={ON|OFF|*period*}] [FRAGMENT={ON|OFF}]
[FRAGOVERHEAD=0..100] [IDLE={ON|OFF|*time*}]
[INDATALIMIT={NONE|1..65535}] [IPPOOL={*pool-name*|NONE}]
[IPREQUEST={ON|OFF}] [LOGIN=USER] [LQR={ON|OFF|*time*}]
[MAGIC={ON|OFF}] [MAXLINKS=1..64] [MULTILINK={ON|OFF}]
[NULLFRAGTIMER=*time*] [ONLINELIMIT={NONE|1..65535}]
[OUTDATALIMIT={NONE|1..65535}] [PASSWORD=*password*]
[RESTART=*time*] [STACHECK={LCB|SEQUENCE}]
[TOTALDATALIMIT={NONE|1..65535}] [USERNAME=*username*]

where:

- *template* is a number in the range 0 to 31.
- *e164number* is the phone number to dial when performing callback. It may contain digits (0–9) and should be a valid phone number as described in CCITT standard E.164.
- *description* is a character string, 1 to 70 characters in length. Valid characters are any printable character.
- *period* is a decimal number in the range 1 to 4294967295.
- *time* is a timer value in seconds.
- *pool-name* is a character string, 1 to 15 characters in length. Valid characters are any printable characters. If *pool-name* contains spaces, it must be enclosed in double quotes.
- *password* is the password to use for authentication, 1 to 32 characters in length. It may contain any printable character, and is case sensitive.
- *username* is the username to use for authentication, 1 to 32 characters in length. It may contain any printable character, and is case sensitive.

Description This command creates a PPP template that is used to configure dynamic PPP interfaces that are created when an ISDN call or PPP over Ethernet service is activated.

The TEMPLATE parameter specifies the number of the template to create. The specified template must not already exist.

The AUTHENTICATION parameter specifies the authentication protocol to be used on the physical interface or channel. If CHAP is specified, the Challenge-Handshake Authentication Protocol (CHAP) is used. If PAP is specified, the Password Authentication Protocol (PAP) is used. If EITHER is specified, the router uses the option negotiation process to negotiate the authentication protocol to be used with the device at the remote end of the link, specifying CHAP as the first choice. If NONE is specified, no authentication protocol is used. The default is NONE.

The BAP parameter specifies whether or not the Bandwidth Allocation Protocol will be used for negotiating the activation of demand PPP links. The default is ON.

The BAPMODE parameter specifies which peer originates another link to add to the multilink bundle. For CALLBACK mode, the number to call must be configured on the call at the lower layer (ISDN). The default is CALL.

The CBDELAY parameter specifies the delay, in tenths of a second, between bringing down a call for callback and actually making the call back to the peer. This parameter is used to handle the different timing requirements of various ISDN switches and is only valid for PPP links over ISDN calls and when the callback mode is REQUEST. The default is 1.

The CBMODE parameter specifies whether a callback request will be made or accepted during the LCP negotiation. If REQUEST is specified a request will be made for callback. If ACCEPT is specified requests for callback received from the peer will be accepted and processed, however AUTHENTICATION must be set to PAP or CHAP. If OFF is specified no callback requests will be made and callback requests will not be accepted. The default is OFF.

The CBNUMBER parameter specifies the number to include when requesting a callback with the CBOperation parameter set to E164NUMBER. The number specified should be a phone number as specified in the E.164 standard.

The CBOperation parameter specifies the callback operation to be included in the callback request to specify to the peer how to determine the callback number. If USERAUTH is specified the peer will use the username and password supplied during authentication to look up the callback number. If E164NUMBER is specified the callback number specified by the CBNUMBER parameter is included in the callback request. The default is USERAUTH.

The COMPALGORITHM parameter specifies the compression algorithm to use when compressing and decompressing PPP packets. If STACLZS is specified the Stac LZS algorithm will be used as specified in RFC 1974. The default is STACLZS.

The COMPRESSION parameter enables or disables the use of compression for the interface. When used with multilink, setting COMPRESSION to ON will compress the packets before they are sent to the individual links. Setting COMPRESSION to LINK will enable compression for the link specified by the OVER parameter. The default is OFF. The LINK option should only be used when compression is required on some physical interfaces and not on others. If

compression is required on all physical interfaces of a PPP interface, the `COMPRESSION` parameter should be set to `ON`.

The `COPY` parameter specifies the name of an existing template to copy as the default values for this template. Any other parameters modify the copy.

The `DEBUGMAXBYTES` parameter specifies the maximum number of bytes that are displayed for each packet when the `PACKET` debug option is enabled. The default is 32.

The `DESCRIPTION` parameter specifies a user-defined description for the interface, to make it easier to distinguish between a number of PPP interfaces.

The `ECHO` parameter specifies whether or not LCP *Echo Request* and *Echo Reply* messages are used to determine link quality. If three consecutive *Echo Request* messages are transmitted without receiving an *Echo Reply* response, the link is deemed to be down. The `ECHO` and `LQR` parameters are mutually exclusive. If `ECHO` is enabled, `LQR` will be disabled. If `LQR` is enabled, `ECHO` will be set to `OFF`. If `OFF` is specified, *Echo Request* messages will not be transmitted. If `ON` is specified, *Echo Request* messages will be transmitted every 60 seconds. If a period in seconds is specified, *Echo Request* messages are transmitted at the specified interval.

The `FRAGMENT` parameter applies only to a multilink bundle interface, and determines whether packets are fragmented or not. The default is `OFF`. Fragmentation must be disabled if compression is required.

The `FRAGOVERHEAD` parameter specifies the maximum allowable overhead, as a percentage, for fragmenting packets using the variable fragmentation scheme for multilink PPP. If this limit will be exceeded for any packet the packet is fragmented using the fixed fragmentation scheme. The default is 5. The variable fragmentation scheme spreads the packet over all the links in the multilink bundle by splitting the packet into variable sized fragments to match the speed of individual links. Larger fragments are transmitted over faster links, thereby providing an inherent load balancing scheme. The fixed fragmentation scheme spreads the packet over all the links in the multilink bundle by splitting the packet into equal fixed sized fragments. If the number of links is large and the packet is relatively small a fragment is not transmitted over every link.

The `IDLE` parameter controls the dial-on-demand feature. If `ON` is specified, dial-on-demand is enabled with a default timer of 60 seconds. If a time is specified, dial-on-demand is enabled with the timer set to the specified time. If `OFF` is specified, dial-on-demand is disabled. When the dial-on-demand feature is activated, PPP brings up the link when there is traffic to be sent, and takes down the link when there has been no traffic for the specified timer period. The effect on a PPP interface using an ISDN call will be to connect the call when traffic is to be sent and disconnect the call when no traffic has been sent or received for the specified timer period. For other physical interfaces, this parameter has no effect, as the links are always connected. The default is `OFF`.

The `INDATALIMIT` parameter specifies the input data threshold, in megabytes, for the PPP interface. When the interface's cumulative input data counter exceeds this limit, the link is closed and any further attempts to open the link will fail. If the input data counter for the interface is cleared using the `RESET PPP` command on page 3-48, the link can be reopened. The default is `NONE`, which sets no threshold.

The IPPOOL parameter specifies the IP pool to use to allocate IP addresses for dynamic dial-in PPP connections. If NONE is specified, IP addresses are not allocated from an IP pool. The default is NONE. See “IP Address Pools” on page 6-32 of *Chapter 6, Internet Protocol (IP)* for more information about creating IP address pools.

The IPREQUEST parameter specifies whether or not a request will be made for an IP address to be allocated by the peer during the IPCP negotiation. If ON is specified a request will be made. If OFF is specified a request will not be specified. The default is OFF.

The LOGIN parameter specifies which login procedure the call creating this dynamic interface must use when it is activated. If USER is specified, the router will check the User Authentication Database to authenticate the call.

The LQR parameter sets the LQR timer. If ON is specified, LQR is enabled with a default timer of 60 seconds. If a time is specified, LQR is enabled with the timer set to the specified time. If OFF is specified, LQR is disabled. The default is ON.

The MAGIC parameter enables or disables negotiation of the magic number option. The default is ON. The magic number is used to determine if a interface is looped back. The interface will not reach the OPENED state if there is a loopback.

The MAXLINKS parameter specifies the maximum number of links allowed in a multilink PPP interface created using this template.

The MULTILINK parameter specifies whether or not incoming dynamic PPP calls using this template will be multilinked together. If ON is specified, incoming dynamic PPP calls from the same peer and with the same authentication information will be multilinked together. If OFF is specified, incoming dynamic PPP calls will not be multilinked under any circumstances. The default is ON.

The NULLFRAGTIMER parameter specifies the maximum time, in seconds, a link in a multilink bundle may be idle before a NULL fragment is transmitted over the link. NULL fragments are used to keep the last sequence number transmitted over the link up to date. The default is 3.

The ONLINELIMIT parameter specifies the up-time threshold, in hours, for the PPP interface. When the interface’s cumulative up-time counter exceeds this limit, the link is closed and any further attempts to open the link will fail. If the up-time counter for the interface is cleared using the RESET PPP command on page 3-48, the link can be reopened. The default is NONE, which sets no threshold.

The OUTDATALIMIT parameter specifies the output data threshold, in megabytes, for the PPP interface. When the interface’s cumulative output data counter exceeds this limit, any further attempts to open the link will fail. If the output data counter for the interface is cleared using the RESET PPP command on page 3-48, the link can be reopened. The default is NONE, which sets no threshold.

The PASSWORD parameter specifies the password to use when the peer requests authentication using either CHAP or PAP. This is normally required for network lines between routers, for which an authentication protocol has been selected with the AUTHENTICATION parameter.

The RESTART parameter specifies the time between successive retransmissions of unacknowledged configure requests or terminate requests. The default is 3 seconds.

The STACCHECK parameter specifies the check mode to used for the Stac LZS compression algorithm. If SEQUENCE is specified an incrementing sequence number is used to determine whether a packet has been lost and therefore whether the compression history needs to be reset. If LCB is specified an LCB value is used to determine if an error has occurred in a packet. The default is SEQUENCE.

The TOTALDATALIMIT parameter sets the total data throughput threshold, in megabytes, for the PPP interface. When the interface's cumulative total data counter exceeds this limit, the link is closed and any further attempts to open the link will fail. If the total data counter for the interface is cleared using the RESET PPP command on page 3-48, the link can be reopened. The default is NONE which corresponds to no threshold.

The USERNAME parameter specifies the username to be used when generating PAP authentication requests and when responding to CHAP authentication challenges. If the USERNAME is not set the router's system name will be used by default.



For security reasons this command will only be accepted if the user has SECURITY OFFICER privilege.

Examples To create a template that creates a dynamic PPP interface with BAP and STAC LZS compression enabled, use the command:

```
CREATE PPP TEMPLATE=1 BAP=ON BAPMODE=CALL
DESCRIPTION="Dynamic PPP with STAC LZS and BAP"
COMPRESSION=ON COMPALGORITHM=STACLZS
```

To create PPP template pppT2 using the factory default settings, use the command:

```
CREATE PPP TEMPLATE=2
```

See Also DESTROY PPP TEMPLATE
DISABLE PPP TEMPLATE DEBUG
ENABLE PPP TEMPLATE DEBUG
RESET PPP
SET PPP TEMPLATE
SHOW PPP TEMPLATE
SHOW PPP LIMITS

DELETE PPP

Syntax DELETE PPP=*ppp-interface* OVER=*physical-interface*
[NUMBER=*number*] [TYPE={DEMAND|PRIMARY|SECONDARY}]

where:

- *ppp-interface* is the PPP interface number.
- *physical-interface* is ISDN-callname, MIOX*n*-circuitname, TDM-groupname or ETH*n*-servicename.
- *number* is the number of physical interfaces to delete.

Description This command deletes the specified ISDN call, MIOX circuit, TDM group or PPP over Ethernet service from use by a PPP interface as a physical layer. The interface may be left with no physical layers.

The OVER parameter specifies the physical interface to be deleted.

The NUMBER parameter specifies the number of physical interfaces to be deleted. This parameter is only valid when the OVER parameter specifies an ISDN call as the physical interface. The default is 1.

The TYPE parameter specifies the role of the physical interface for bandwidth on demand and leased line backup. The default is PRIMARY.

Examples To delete ISDN call “demand” as a physical interface from PPP interface 1, use the command:

```
DELETE PPP=1 OVER=ISDN-demand
```

See Also ADD PPP
CREATE PPP
DESTROY PPP
DISABLE PPP
ENABLE PPP
RESET PPP
SET PPP
SHOW PPP

DESTROY PPP

Syntax DESTROY PPP=*ppp-interface*

where:

- *ppp-interface* is the PPP interface number.

Description This command destroys the specified PPP interface, as opposed to the DELETE PPP command on page 3-41 which deletes a physical interface used by a PPP interface.

Examples To destroy PPP interface 0, use the command:

```
DESTROY PPP=0
```

See Also ADD PPP
CREATE PPP
DELETE PPP
DISABLE PPP
ENABLE PPP
RESET PPP
SET PPP
SHOW PPP

DESTROY PPP TEMPLATE

Syntax DESTROY PPP TEMPLATE=*template*

where:

- *template* is a number in the range 0 to 31.

Description This command destroys the specified PPP template and eliminates any call associations. The TEMPLATE parameter specifies the number of the template to destroy. The specified template must already exist.

Examples To destroy template 1, use the command:

```
DESTROY PPP TEMPLATE=1
```

See Also CREATE PPP TEMPLATE
DISABLE PPP TEMPLATE DEBUG
ENABLE PPP TEMPLATE DEBUG
SET PPP TEMPLATE
SHOW PPP TEMPLATE

DISABLE PPP

Syntax DISABLE PPP=*ppp-interface*

where:

- *ppp-interface* is the PPP interface number.

Description This command disables the specified PPP interface. The interface must currently be enabled. The interface is not available for use by higher layer network protocols, but the configuration is retained in nonvolatile storage and is restored when the interface is re-enabled.

Examples To disable PPP interface 2, use the command:

```
DISABLE PPP=2
```

See Also ADD PPP
 CREATE PPP
 DELETE PPP
 DISABLE PPP
 ENABLE PPP
 RESET PPP
 SET PPP
 SHOW PPP

DISABLE PPP DEBUG

Syntax `DISABLE PPP=ppp-interface DEBUG={ALL|AUTH|BAPPKT|BAPSTATE|CALLBACK|DEMAND|ENCO|LCP|NCP|PKT|UTILISATION} [, ...]`

where:

■ *ppp-interface* is the PPP interface number.

Description This command disables the debugging option for the specified PPP interface. The option must currently be enabled. A list of options separated by commas may be specified to disable more than one debugging option at a time.

The DEBUG parameter specifies which debugging options are to be disabled. The value of this parameter is a single item or a comma-separated list of items. The items allowed and the debugging that results from specifying the item are shown in Table 3-4 on page 3-45.

Examples To disable all debugging options on PPP interface 2, use the command:

```
DISABLE PPP=2 DEBUG=ALL
```

See Also ENABLE PPP DEBUG

DISABLE PPP TEMPLATE DEBUG

Syntax `DISABLE PPP TEMPLATE=template DEBUG={ALL|AUTH|BAPPKT|BAPSTATE|CALLBACK|DEMAND|ENCO|LCP|NCP|PKT|UTILISATION} [, ...]`

where:

■ *template* is a number in the range 0 to 31.

Description This command disables the debugging option for dynamic PPP interfaces created using the specified PPP template. A list of options separated by commas may be specified to disable more than one debugging option at a time.

The TEMPLATE parameter specifies the number of the template for which debugging is to be disabled. The specified template must already exist.

The DEBUG parameter specifies which debugging options are to be disabled. The value of this parameter is a single item or a comma-separated list of items.

The items allowed and the debugging that results from specifying the item are shown in Table 3-4 on page 3-45.

Examples To disable the display of debugging information for dial on demand link activation on template 2, use the command:

```
DISABLE PPP TEMPLATE=2 DEBUG=DEMAND
```

See Also CREATE PPP TEMPLATE
DESTROY PPP TEMPLATE
ENABLE PPP TEMPLATE DEBUG
SET PPP TEMPLATE
SHOW PPP TEMPLATE

ENABLE PPP

Syntax `ENABLE PPP=ppp-interface`

where:

■ *ppp-interface* is the PPP interface number.

Description This command enables the specified PPP interface. The interface must currently be disabled. The interface configuration is restored to the settings in existence before the interface was disabled. The interface is made available to network layer protocols to transmit and receive data.

Examples To enable PPP interface 2, use the command:

```
ENABLE PPP=2
```

See Also ADD PPP
CREATE PPP
DELETE PPP
DESTROY PPP
DISABLE PPP
RESET PPP
SET PPP
SHOW PPP

ENABLE PPP DEBUG

Syntax `ENABLE PPP=ppp-interface DEBUG={ALL|AUTH|BAPPKT|BAPSTATE|CALLBACK|DEMAND|ENCO|LCP|NCP|PKT|UTILISATION} [, ...]
[PORT=port-number] [TIMEOUT={NONE|1..4000000000}]
[NUMPKTS={CONT|1..4000000000}]`

where:

- *ppp-interface* is the PPP interface number.
- *port-number* is the number of an asynchronous port on the router. Ports are numbered starting at zero (0).

Description This command enables the debugging option for the specified PPP interface. Debugging may or may not be enabled already. Debugging information is sent to the port or telnet session from which the command was entered if the PORT parameter was not specified, otherwise it is sent to the specified port. A list of options separated by commas may be specified to enable more than one debugging option at a time. For packet debugging, the number of packets output may be specified. For all other types of debugging, the length of time that debugging will continue may be specified.

The DEBUG parameter specifies which debugging options are to be disabled. The value of this parameter is a single item or a comma-separated list of items. The items allowed and the debugging that results from specifying the item are shown in Table 3-4 on page 3-45.

Table 3-4: Point-to-Point Protocol (PPP) debugging options.

Option	Description
ALL	All debug options.
AUTH	PPP authentication. If LCP opens on a link but the network protocols remain in the CLOSED state, the most likely cause is an authentication failure.
BAPPKT	BAP packets received over the interface.
BAPSTATE	BAP state machine transitions.
CALLBACK	Callback state machine transitions.
DEMAND	Packets that cause on-demand links to be activated.
ENCO	ENCO state machine used to control attachment to and detachment from the ENCO (compression) module.
LCP	LCP state machine transitions.
NCP	NCP state machine transitions.
PKT	All packets received and transmitted on the PPP interface.
UTILISATION	Utilisation measurements for each lower layer interface and the overall utilisation.



Enabling all debug options with `ENABLE PPP DEBUG=ALL` may generate enormous amounts of output, causing the router to lock up. Use the `TIMEOUT` or `NUMPKTS` options to limit the amount of output generated.

The PORT parameter specifies the asynchronous port to which the debug output is to be sent. This enables debugging to be enabled in a script. The

default is to send the output to the terminal or Telnet session from which the command was executed. Each time the ENABLE PPP DEBUG command is entered the destination of the debugging output is calculated again using this rule.

The TIMEOUT parameter specifies a time in seconds after which debugging will automatically cease. If NONE is specified then debugging must be disabled manually. The timeout only applies to debugging modes which do not involve the output of data packets, that is, all debugging modes except for PKT. The value of the TIMEOUT parameter the first time an applicable debugging mode is enabled will be retained for future ENABLE PPP DEBUG commands. The default is NONE.

The NUMPKTS parameter specifies, for PKT debugging, the number of packets to be displayed before debugging ceases. This option is useful when attempting to debug a very busy link, since the amount of output generated by PKT debugging can easily cause the router to lock up the device to which the debugging output is being sent. The value of this parameter the first time PKT debugging is enabled will be retained for subsequent ENABLE PPP DEBUG commands. If CONT is specified, packet debugging will continue indefinitely and must be disabled manually. The default is CONT.

Examples To enable the display of debugging information for dial on demand link activation on PPP interface 2, use the command:

```
ENABLE PPP=2 DEBUG=DEMAND
```

See Also DISABLE PPP DEBUG

ENABLE PPP TEMPLATE DEBUG

Syntax `ENABLE PPP TEMPLATE=template DEBUG={ALL|AUTH|BAPPKT|BAPSTATE|CALLBACK|DEMAND|ENCO|LCP|NCP|PKT|UTILISATION}[,...] [PORT=port-number] [TIMEOUT={NONE|1..4000000000}] [NUMPKTS={CONT|1..4000000000}]`

where:

- *template* is a number in the range 0 to 31.
- *port-number* is the number of an asynchronous port on the router. Ports are numbered starting at zero (0).

Description This command enables the debugging option for dynamic PPP interfaces created using the specified PPP template. Debugging may or may not be enabled already. Debugging information is sent to the port or telnet session from which the command was entered if the PORT parameter was not specified, otherwise it is sent to the specified port. A list of options separated by commas may be specified to enable more than one debugging option at a time. For packet debugging, the number of packets output may be specified. For all other types of debugging, the length of time that debugging will continue may be specified.

The TEMPLATE parameter specifies the number of the template for which debugging is to be enabled. The specified template must already exist.

The DEBUG parameter specifies which debugging options are to be disabled. The value of this parameter is a single item or a comma-separated list of items. The items allowed and the debugging that results from specifying the item are shown in Table 3-4 on page 3-45.



Enabling all debug options with ENABLE PPP TEMPLATE DEBUG=ALL may generate enormous amounts of output, causing the router to lock up. Use the TIMEOUT or NUNPKTS options to limit the amount of output generated.

The PORT parameter specifies the asynchronous port to which the debug output is to be sent. This enables debugging to be enabled in a script. The default is to send the output to the terminal or Telnet session from which the command was executed. Each time the ENABLE PPP TEMPLATE DEBUG command is entered the destination of the debugging output is calculated again using this rule.

The TIMEOUT parameter specifies a time in seconds after which debugging will automatically cease. If NONE is specified then debugging must be disabled manually. The timeout only applies to debugging modes which do not involve the output of data packets, that is, all debugging modes except for PKT. The value of the TIMEOUT parameter the first time an applicable debugging mode is enabled will be retained for future ENABLE PPP TEMPLATE DEBUG commands. The default is NONE.

The NUNPKTS parameter specifies, for PKT debugging, the number of packets to be displayed before debugging ceases. This option is useful when attempting to debug a very busy link, since the amount of output generated by PKT debugging can easily cause the router to lock up the device to which the debugging output is being sent. The value of this parameter the first time PKT debugging is enabled will be retained for subsequent ENABLE PPP TEMPLATE DEBUG commands. If CONT is specified, packet debugging will continue indefinitely and must be disabled manually. The default is CONT.

Examples To enable the display of debugging information for dial on demand link activation on template 2, use the command:

```
ENABLE PPP TEMPLATE=2 DEBUG=DEMAND
```

See Also CREATE PPP TEMPLATE
DESTROY PPP TEMPLATE
DISABLE PPP TEMPLATE DEBUG
SET PPP TEMPLATE
SHOW PPP TEMPLATE

PURGE PPP

Syntax PURGE PPP

Description This command destroys all PPP interfaces and reinitialises the PPP module.

Examples To the PPP configuration, use the command:

```
PURGE PPP
```

See Also DELETE PPP
DESTROY PPP
DISABLE PPP
ENABLE PPP

RESET PPP

Syntax RESET PPP=*ppp-interface* [COUNTERS] [LINKCOUNTERS={ONLINE |
INDATA | OUTDATA | TOTALDATA | ALL}]

where:

■ *ppp-interface* is the PPP interface number.

Description This command resets the specified PPP interface, or the counters for the specified PPP interface. This command resets the specified PPP interface, the general counters for the specified PPP interface, or the cumulative up-time, and input/output data counters.

If the COUNTERS parameter is specified, all counters for the interface are reset to zero (0) except for the cumulative up-time, and input/output data counters.

If the LINKCOUNTERS parameter is specified, one or all of the up-time and input/output data counters are reset to zero (0). If ONLINE is specified the up-time counter is reset to zero. If INDATA is specified the input data counter is reset to zero. If OUTDATA is specified the output data counter is reset to zero. If OUTDATA is specified the total data counter is reset to zero. If ALL is specified all four counters are reset to zero.

If neither the COUNTERS nor LINKCOUNTERS parameters are specified, the interface is reset, forcing the interface to renegotiate all protocols and options.

Examples To reset PPP interface 0, use the command:

```
RESET PPP=0
```

To reset all counters for PPP interface 0 without resetting the interface itself, use the command:

```
RESET PPP=0 COUNTERS
```

See Also CREATE PPP
DELETE PPP
DESTROY PPP
DISABLE PPP
ENABLE PPP
PURGE PPP
SET PPP
SHOW PPP LIMITS

SET PPP

Syntax SET PPP [=ppp-interface] [OVER=physical-interface]
 [AUTHENTICATION={CHAP|EITHER|PAP|NONE}] [AUTHMODE={IN|OUT|INOUT}] [BAP={ON|OFF}] [BAPMODE={CALL|CALLBACK}]
 [CBDELAY=1..100] [CBMODE={ACCEPT|OFF|REQUEST}]
 [CBNUMBER=e164number] [CBOperation={E164NUMBER|USERAUTH}] [COMPALGORITHM=STACLSZS] [COMPRESSION={ON|OFF|LINK}] [CONFIGURE={value|CONTINUOUS}]
 [DEBUGMAXBYTES=16..256] [DESCRIPTION=description]
 [DNSPRIMARY=ipadd] [DNSSECONDARY=ipadd]
 [DOWNRATE=0..100] [DOWNTIME=time] [ECHO={ON|OFF|period}] [FRAGMENT={ON|OFF}] [FRAGOVERHEAD=0.100]
 [IDLE={ON|OFF|time}] [INDATALIMIT={NONE|1..65535}]
 [IPPOOL={pool-name|NONE}] [IPREQUEST={ON|OFF}]
 [LQR={ON|OFF|time}] [MAGIC={ON|OFF}]
 [NULLFRAGTIMER=time] [NUMBER=number]
 [ONLINELIMIT={NONE|1..65535}] [OUTDATALIMIT={NONE|1..65535}] [PASSWORD=password] [RESTART=time]
 [STACHECK={LCB|SEQUENCE}] [TERMINATE={value|CONTINUOUS}] [TOTALDATALIMIT={NONE|1..65535}]
 [TYPE={DEMAND|PRIMARY|SECONDARY}] [UPRATE=0..100]
 [UPTIME=time] [USERNAME=username] [WINSPRIMARY=ipadd]
 [WINSSECONDARY=ipadd]

where:

- *ppp-interface* is the PPP interface number.
- *physical-interface* is ISDN-callname, MIOXn-circuitname, TDM-groupname or ETHn-servicename.
- *e164number* is the phone number to dial when performing callback. It may contain digits (0–9) and should be a valid phone number as described in CCITT standard E.164.
- *value* is a retry threshold.
- *ipadd* is an IP address in dotted decimal notation.
- *description* is a character string, 1 to 70 characters in length. Valid characters are any printable character.
- *time* is a timer value in seconds.
- *period* is a decimal number in the range 1 to 4294967295.
- *pool-name* is a character string, 1 to 15 characters in length. Valid characters are any printable characters. If *pool-name* contains spaces, it must be enclosed in double quotes.
- *number* is the number of PPP interfaces to create.
- *password* is the password to use for authentication, 1 to 32 characters in length. It may contain any printable character, and is case sensitive.
- *username* is the username to use for authentication, 1 to 32 characters in length. It may contain any printable character, and is case sensitive.

Description This command is used to change the configuration parameters of a PPP interface running over an ISDN call, a MIOX circuit, a TDM group or a PPP over Ethernet service (referred to as a physical layer). It is also used to set global primary and secondary DNS and WINS server addresses. In this case

the PPP interface may not be specified. All other options require the PPP interface to be specified.

The OVER parameter specifies the physical interface over which the PPP interface is running.

The AUTHENTICATION parameter specifies the authentication protocol to be used on the physical interface or channel. If CHAP is specified, the Challenge-Handshake Authentication Protocol (CHAP) is used. If PAP is specified, the Password Authentication Protocol (PAP) is used. If EITHER is specified, the router uses the option negotiation process to negotiate the authentication protocol to be used with the device at the remote end of the link, specifying CHAP as the first choice. If NONE is specified, no authentication protocol is used. The default is NONE.

The AUTHMODE parameter specifies how authentication requests to peers are affected by the direction of the ISDN call. The AUTHMODE parameter is only valid when the AUTHENTICATION parameter is set to a value other than NONE and the physical interface is an ISDN call. If IN is specified, authentication will only be requested for incoming calls from peers. If OUT is specified, authentication will only be requested for outgoing calls to peers. If INOUT is specified, authentication will always be requested regardless of the direction of the call. The default is INOUT.

The BAP parameter specifies whether or not the Bandwidth Allocation Protocol will be used for negotiating the activation of demand PPP links. The default is ON.

The BAPMODE parameter specifies which peer originates another link to add to the multilink bundle. For CALLBACK mode, the number to call must be configured on the call at the lower layer (ISDN). The default is CALL.

The CBDELAY parameter specifies the delay, in tenths of a second, between bringing down a call for callback and actually making the call back to the peer. This parameter is used to handle the different timing requirements of various ISDN switches and is only valid for PPP links over ISDN calls and when the callback mode is REQUEST. The default is 1.

The CBMODE parameter specifies whether a callback request will be made or accepted during the LCP negotiation. If REQUEST is specified a request will be made for callback. If ACCEPT is specified requests for callback received from the peer will be accepted and processed, however AUTHENTICATION must be set to PAP or CHAP. If OFF is specified no callback requests will be made and callback requests will not be accepted. The default is OFF.

The CBNUMBER parameter specifies the number to include when requesting a callback with the CBOperation parameter set to E164NUMBER. The number specified should be a phone number as specified in the E.164 standard.

The CBOperation parameter specifies the callback operation to be included in the callback request to specify to the peer how to determine the callback number. If USERAUTH is specified the peer will use the username and password supplied during authentication to look up the callback number. If E164NUMBER is specified the callback number specified by the CBNUMBER parameter is included in the callback request. The default is USERAUTH.

The COMPALGORITHM parameter specifies the compression algorithm to use when compressing and decompressing PPP packets. If STACLS is specified

the Stac LZS algorithm will be used as specified in RFC 1974. The default is STACLZS.

The COMPRESSION parameter enables or disables the use of compression for the interface. When used with multilink, setting COMPRESSION to ON will compress the packets before they are sent to the individual links. Setting COMPRESSION to LINK will enable compression for the link specified by the OVER parameter. The default is OFF. The LINK option should only be used when compression is required on some physical interfaces and not on others. If compression is required on all physical interfaces of a PPP interface, the COMPRESSION parameter should be set to ON.

The CONFIGURE parameter sets the number of configure requests sent before some action is taken. For the LCP the action is to reset the hardware and start again. For all other protocols the action is to give up. The default is CONTINUOUS, which means that requests will be sent continuously.

The DEBUGMAXBYTES parameter specifies the maximum number of bytes that are displayed for each packet when the PACKET debug option is enabled. The default is 32.

The DESCRIPTION parameter specifies a user-defined description for the interface, to make it easier to distinguish between a number of PPP interfaces.

The DNSPRIMARY parameter specifies the IP address to pass to a peer when it requests a primary DNS address using the IPCP primary DNS option.

The DNSSECONDARY parameter specifies the IP address to pass to a peer when it requests a secondary DNS address using the IPCP secondary DNS option.

The DOWNTIME parameter specifies the time, in seconds, that the PPP interface must have a total utilisation (as a percentage) below the threshold specified by the DOWNRATE parameter, before a channel is closed. The default is 60 for DOWNTIME and 20 for DOWNRATE. The UPRATE, UPTIME, DOWNRATE and DOWNTIME parameters are used in conjunction with the TYPE parameter to configure bandwidth on demand.

The ECHO parameter specifies whether or not LCP *Echo Request* and *Echo Reply* messages are used to determine link quality. If three consecutive *Echo Request* messages are transmitted without receiving an *Echo Reply* response, the link is deemed to be down. The ECHO and LQR parameters are mutually exclusive. If ECHO is enabled, LQR will be disabled. If LQR is enabled, ECHO will be set to OFF. If OFF is specified, *Echo Request* messages will not be transmitted. If ON is specified, *Echo Request* messages will be transmitted every 60 seconds. If a period in seconds is specified, *Echo Request* messages are transmitted at the specified interval.

The FRAGMENT parameter applies only to a multilink bundle interface, and determines whether packets are fragmented or not. The default is OFF. Fragmentation must be disabled if compression is required.

The FRAGOVERHEAD parameter specifies the maximum allowable overhead, as a percentage, for fragmenting packets using the variable fragmentation scheme for multilink PPP. If this limit will be exceeded for any packet the packet is fragmented using the fixed fragmentation scheme. The default is 5. The variable fragmentation scheme spreads the packet over all the links in the multilink bundle by splitting the packet into variable sized fragments to match the speed of individual links. Larger fragments are transmitted over faster

links, thereby providing an inherent load balancing scheme. The fixed fragmentation scheme spreads the packet over all the links in the multilink bundle by splitting the packet into equal fixed sized fragments. If the number of links is large and the packet is relatively small a fragment is not transmitted over every link.

The IDLE parameter controls the dial-on-demand feature. If ON is specified, dial-on-demand is enabled with a default timer of 60 seconds. If a time is specified, dial-on-demand is enabled with the timer set to the specified time. If OFF is specified, dial-on-demand is disabled. When the dial-on-demand feature is activated, PPP brings up the link when there is traffic to be sent, and takes down the link when there has been no traffic for the specified timer period. The effect on a PPP interface using an ISDN call will be to connect the call when traffic is to be sent and disconnect the call when no traffic has been sent or received for the specified timer period. For other physical interfaces, this parameter has no effect, as the links are always connected. The default is OFF.

The INDATALIMIT parameter specifies the input data threshold, in megabytes, for the PPP interface. When the interface's cumulative input data counter exceeds this limit, the link is closed and any further attempts to open the link will fail. If the input data counter for the interface is cleared using the RESET PPP command on page 3-48, the link can be reopened. The default is NONE, which sets no threshold.

The IPOOL parameter specifies the IP pool to use to allocate IP addresses for the remote end of the PPP connection. If NONE is specified, IP addresses are not allocated from an IP pool. The default is NONE. See "IP Address Pools" on page 6-32 of *Chapter 6, Internet Protocol (IP)* for more information about creating IP address pools.

The IPREQUEST parameter specifies whether or not a request will be made for an IP address to be allocated by the peer during the IPCP negotiation. If ON is specified a request will be made. If OFF is specified a request will not be specified. The default is OFF.

The LQR parameter sets the LQR timer. If ON is specified, LQR is enabled with a default timer of 60 seconds. If a time is specified, LQR is enabled with the timer set to the specified time. If OFF is specified, LQR is disabled. The default is ON.

The MAGIC parameter enables or disables negotiation of the magic number option. The default is ON. The magic number is used to determine if a interface is looped back. The interface will not reach the OPENED state if there is a loopback.

The NUMBER parameter specifies the number of physical interfaces to be created. This parameter is only valid when the OVER parameter specifies an ISDN call as the physical interface. The default is 1.

The NULLFRAGTIMER parameter specifies the maximum time, in seconds, a link in a multilink bundle may be idle before a NULL fragment is transmitted over the link. NULL fragments are used to keep the last sequence number transmitted over the link up to date. The default is 3.

The ONLINELIMIT parameter specifies the up-time threshold, in hours, for the PPP interface. When the interface's cumulative up-time counter exceeds this limit, the link is closed and any further attempts to open the link will fail. If the up-time counter for the interface is cleared using the RESET PPP command on

page 3-48, the link can be reopened. The default is NONE, which sets no threshold.

The OUTDATALIMIT parameter specifies the output data threshold, in megabytes, for the PPP interface. When the interface's cumulative output data counter exceeds this limit, any further attempts to open the link will fail. If the output data counter for the interface is cleared using the RESET PPP command on page 3-48, the link can be reopened. The default is NONE, which sets no threshold.

The PASSWORD parameter specifies the password to use when the peer requests authentication using either CHAP or PAP. This is normally required for network lines between routers, for which an authentication protocol has been selected with the AUTHENTICATION parameter.

The RESTART parameter specifies the time between successive retransmissions of unacknowledged configure requests or terminate requests. The default is 3 seconds.

The STACCHECK parameter specifies the check mode to used for the Stac LZS compression algorithm. If SEQUENCE is specified an incrementing sequence number is used to determine whether a packet has been lost and therefore whether the compression history needs to be reset. If LCB is specified an LCB value is used to determine if an error has occurred in a packet. The default is SEQUENCE.

The TERMINATE parameter sets the number of terminate requests sent when trying to close a link before it is assumed the link is down. The default is 2. The CONTINUOUS option specifies that requests will be sent continuously.

The TOTALDATA LIMIT parameter sets the total data throughput threshold, in megabytes, for the PPP interface. When the interface's cumulative total data counter exceeds this limit, the link is closed and any further attempts to open the link will fail. If the total data counter for the interface is cleared using the RESET PPP command on page 3-48, the link can be reopened. The default is NONE which corresponds to no threshold.

The TYPE parameter specifies the role of the physical interface for bandwidth on demand and leased line backup. The default is PRIMARY. If PRIMARY is specified, the link will be kept open all the time (IDLE=OFF) or opened whenever there is traffic (IDLE=ON). If SECONDARY is specified, the link will be opened only when the associated primary link fails. If DEMAND is specified, the link will be opened only when the additional bandwidth is required.

To configure bandwidth on demand the ISDN channels are given a TYPE of DEMAND when they are added to the PPP interface. Adding one channel with a type of PRIMARY and other channels with a TYPE of DEMAND will ensure that there is always one channel available. If all channels are assigned a TYPE of DEMAND then there will be no channels open when there is no traffic, some traffic will cause one channel to be opened and continuous traffic will cause other channels to be opened. If there is one channel remaining opened then the IDLE timer is used to determine when this channel should be closed. In this case it is recommended that the IDLE timer not be set to OFF.

The UPTIME parameter specifies the time, in seconds, that the PPP interface must have a total utilisation (as a percentage) above the threshold specified by the UPRATE parameter, before an additional channel is opened. The default is 30 for UPTIME and 80 for UPRATE. The UPRATE, UPTIME, DOWNRATE and

DOWNTIME parameters are used in conjunction with the TYPE parameter to configure bandwidth on demand.

The USERNAME parameter specifies the username to be used when generating PAP authentication requests and when responding to CHAP authentication challenges. If the USERNAME is not set the router's system name will be used by default.

The WINSPRIMARY parameter specifies the IP address to pass to a peer when it requests a primary WINS server address using the IPCP primary WINS server option.

The WINSSECONDARY parameter specifies the IP address to pass to a peer when it requests a primary WINS server address using the IPCP secondary WINS server option.

Examples To disable compression on PPP interface 1, use the command:

```
SET PPP=1 COMP=OFF
```

See Also ADD PPP
CREATE PPP
RESET PPP
SHOW PPP
SHOW PPP LIMITS

SET PPP TEMPLATE

Syntax SET PPP TEMPLATE=*template* [AUTHENTICATION={CHAP|EITHER|PAP|NONE}] [BAP={ON|OFF}] [BAPMODE={CALL|CALLBACK}] [CBDELAY=1..100] [CBMODE={ACCEPT|OFF|REQUEST}] [CBNUMBER=*e164number*] [CBOperation={E164NUMBER|USERAUTH}] [COMPALGORITHM=STACLSZS] [COMPRESSION={ON|OFF|LINK}] [DEBUGMAXBYTES=16..256] [DESCRIPTION=*description*] [ECHO={ON|OFF|*period*}] [FRAGMENT={ON|OFF}] [FRAGOVERHEAD=0..100] [IDLE={ON|OFF|*time*}] [INDATALIMIT={NONE|1..65535}] [IPPOOL={*pool-name*|NONE}] [IPREQUEST={ON|OFF}] [LOGIN=USER] [LQR={ON|OFF|*time*}] [MAGIC={ON|OFF}] [MAXLINKS=1..64] [MULTILINK={ON|OFF}] [NULLFRAGTIMER=*time*] [ONLINELIMIT={NONE|1..65535}] [OUTDATALIMIT={NONE|1..65535}] [PASSWORD=*password*] [RESTART=*time*] [STACCHECK={LCB|SEQUENCE}] [TOTALDATALIMIT={NONE|1..65535}] [USERNAME=*username*]

where:

- *template* is a number in the range 0 to 31.
- *e164number* is the phone number to dial when performing callback. It may contain digits (0–9) and should be a valid phone number as described in CCITT standard E.164.
- *description* is a character string, 1 to 70 characters in length. Valid characters are any printable character.
- *period* is a decimal number in the range 1 to 4294967295.

- *time* is a timer value in seconds.
- *pool-name* is a character string, 1 to 15 characters in length. Valid characters are any printable characters. If *pool-name* contains spaces, it must be enclosed in double quotes.
- *password* is the password to use for authentication, 1 to 32 characters in length. It may contain any printable character, and is case sensitive.
- *username* is the username to use for authentication, 1 to 32 characters in length. It may contain any printable character, and is case sensitive.

Description This command modifies an existing PPP template that is used to configure dynamic PPP interfaces that are created when an ISDN call or PPP over Ethernet service is activated.

The TEMPLATE parameter specifies the number of the template to modify. The specified template must already exist.

The AUTHENTICATION parameter specifies the authentication protocol to be used on the physical interface or channel. If CHAP is specified, the Challenge-Handshake Authentication Protocol (CHAP) is used. If PAP is specified, the Password Authentication Protocol (PAP) is used. If EITHER is specified, the router uses the option negotiation process to negotiate the authentication protocol to be used with the device at the remote end of the link, specifying CHAP as the first choice. If NONE is specified, no authentication protocol is used. The default is NONE.

The BAP parameter specifies whether or not the Bandwidth Allocation Protocol will be used for negotiating the activation of demand PPP links. The default is ON.

The BAPMODE parameter specifies which peer originates another link to add to the multilink bundle. For CALLBACK mode, the number to call must be configured on the call at the lower layer (ISDN). The default is CALL.

The CBDELAY parameter specifies the delay, in tenths of a second, between bringing down a call for callback and actually making the call back to the peer. This parameter is used to handle the different timing requirements of various ISDN switches and is only valid for PPP links over ISDN calls and when the callback mode is REQUEST. The default is 1.

The CBMODE parameter specifies whether a callback request will be made or accepted during the LCP negotiation. If REQUEST is specified a request will be made for callback. If ACCEPT is specified requests for callback received from the peer will be accepted and processed, however AUTHENTICATION must be set to PAP or CHAP. If OFF is specified no callback requests will be made and callback requests will not be accepted. The default is OFF.

The CBNUMBER parameter specifies the number to include when requesting a callback with the CBOperation parameter set to E164NUMBER. The number specified should be a phone number as specified in the E.164 standard.

The CBOperation parameter specifies the callback operation to be included in the callback request to specify to the peer how to determine the callback number. If USERAUTH is specified the peer will use the username and password supplied during authentication to look up the callback number. If E164NUMBER is specified the callback number specified by the CBNUMBER parameter is included in the callback request. The default is USERAUTH.

The COMPALGORITHM parameter specifies the compression algorithm to use when compressing and decompressing PPP packets. If STACLZS is specified the Stac LZS algorithm will be used as specified in RFC 1974. The default is STACLZS.

The COMPRESSION parameter enables or disables the use of compression for the interface. When used with multilink, setting COMPRESSION to ON will compress the packets before they are sent to the individual links. Setting COMPRESSION to LINK will enable compression for the link specified by the OVER parameter. The default is OFF. The LINK option should only be used when compression is required on some physical interfaces and not on others. If compression is required on all physical interfaces of a PPP interface, the COMPRESSION parameter should be set to ON.

The DEBUGMAXBYTES parameter specifies the maximum number of bytes that are displayed for each packet when the PACKET debug option is enabled. The default is 32.

The DESCRIPTION parameter specifies a user-defined description for the interface, to make it easier to distinguish between a number of PPP interfaces.

The ECHO parameter specifies whether or not LCP *Echo Request* and *Echo Reply* messages are used to determine link quality. If three consecutive *Echo Request* messages are transmitted without receiving an *Echo Reply* response, the link is deemed to be down. The ECHO and LQR parameters are mutually exclusive. If ECHO is enabled, LQR will be disabled. If LQR is enabled, ECHO will be set to OFF. If OFF is specified, *Echo Request* messages will not be transmitted. If ON is specified, *Echo Request* messages will be transmitted every 60 seconds. If a period in seconds is specified, *Echo Request* messages are transmitted at the specified interval.

The FRAGMENT parameter applies only to a multilink bundle interface, and determines whether packets are fragmented or not. The default is OFF. Fragmentation must be disabled if compression is required.

The FRAGOVERHEAD parameter specifies the maximum allowable overhead, as a percentage, for fragmenting packets using the variable fragmentation scheme for multilink PPP. If this limit will be exceeded for any packet the packet is fragmented using the fixed fragmentation scheme. The default is 5. The variable fragmentation scheme spreads the packet over all the links in the multilink bundle by splitting the packet into variable sized fragments to match the speed of individual links. Larger fragments are transmitted over faster links, thereby providing an inherent load balancing scheme. The fixed fragmentation scheme spreads the packet over all the links in the multilink bundle by splitting the packet into equal fixed sized fragments. If the number of links is large and the packet is relatively small a fragment is not transmitted over every link.

The IDLE parameter controls the dial-on-demand feature. If ON is specified, dial-on-demand is enabled with a default timer of 60 seconds. If a time is specified, dial-on-demand is enabled with the timer set to the specified time. If OFF is specified, dial-on-demand is disabled. When the dial-on-demand feature is activated, PPP brings up the link when there is traffic to be sent, and takes down the link when there has been no traffic for the specified timer period. The effect on a PPP interface using an ISDN call will be to connect the call when traffic is to be sent and disconnect the call when no traffic has been sent or received for the specified timer period. For other physical interfaces, this parameter has no effect, as the links are always connected. The default is OFF.

The INDATALIMIT parameter specifies the input data threshold, in megabytes, for the PPP interface. When the interface's cumulative input data counter exceeds this limit, the link is closed and any further attempts to open the link will fail. If the input data counter for the interface is cleared using the RESET PPP command on page 3-48, the link can be reopened. The default is NONE, which sets no threshold.

The IPOOL parameter specifies the IP pool to use to allocate IP addresses for dynamic dial-in PPP connections. If NONE is specified, IP addresses are not allocated from an IP pool. The default is NONE. See "IP Address Pools" on page 6-32 of *Chapter 6, Internet Protocol (IP)* for more information about creating IP address pools.

The IPREQUEST parameter specifies whether or not a request will be made for an IP address to be allocated by the peer during the IPCP negotiation. If ON is specified a request will be made. If OFF is specified a request will not be specified. The default is OFF.

The LOGIN parameter specifies which login procedure the call creating this dynamic interface must use when it is activated. If USER is specified, the router will check the User Authentication Database to authenticate the call.

The LQR parameter sets the LQR timer. If ON is specified, LQR is enabled with a default timer of 60 seconds. If a time is specified, LQR is enabled with the timer set to the specified time. If OFF is specified, LQR is disabled. The default is ON.

The MAGIC parameter enables or disables negotiation of the magic number option. The default is ON. The magic number is used to determine if a interface is looped back. The interface will not reach the OPENED state if there is a loopback.

The MAXLINKS parameter specifies the maximum number of links allowed in a multilink PPP interface created using this template.

The MULTILINK parameter specifies whether or not incoming dynamic PPP calls using this template will be multilinked together. If ON is specified, incoming dynamic PPP calls from the same peer and with the same authentication information will be multilinked together. If OFF is specified, incoming dynamic PPP calls will not be multilinked under any circumstances. The default is ON.

The NULLFRAGTIMER parameter specifies the maximum time, in seconds, a link in a multilink bundle may be idle before a NULL fragment is transmitted over the link. NULL fragments are used to keep the last sequence number transmitted over the link up to date. The default is 3.

The ONLINELIMIT parameter specifies the up-time threshold, in hours, for the PPP interface. When the interface's cumulative up-time counter exceeds this limit, the link is closed and any further attempts to open the link will fail. If the up-time counter for the interface is cleared using the RESET PPP command on page 3-48, the link can be reopened. The default is NONE, which sets no threshold.

The OUTDATALIMIT parameter specifies the output data threshold, in megabytes, for the PPP interface. When the interface's cumulative output data counter exceeds this limit, any further attempts to open the link will fail. If the output data counter for the interface is cleared using the RESET PPP command

on page 3-48, the link can be reopened. The default is NONE, which sets no threshold.

The PASSWORD parameter specifies the password to use when the peer requests authentication using either CHAP or PAP. This is normally required for network lines between routers, for which an authentication protocol has been selected with the AUTHENTICATION parameter.

The RESTART parameter specifies the time between successive retransmissions of unacknowledged configure requests or terminate requests. The default is 3 seconds.

The STACCHECK parameter specifies the check mode to used for the Stac LZS compression algorithm. If SEQUENCE is specified an incrementing sequence number is used to determine whether a packet has been lost and therefore whether the compression history needs to be reset. If LCB is specified an LCB value is used to determine if an error has occurred in a packet. The default is SEQUENCE.

The TOTALDATALIMIT parameter sets the total data throughput threshold, in megabytes, for the PPP interface. When the interface's cumulative total data counter exceeds this limit, the link is closed and any further attempts to open the link will fail. If the total data counter for the interface is cleared using the RESET PPP command on page 3-48, the link can be reopened. The default is NONE which corresponds to no threshold.

The USERNAME parameter specifies the username to be used when generating PAP authentication requests and when responding to CHAP authentication challenges. If the USERNAME is not set the router's system name will be used by default.

Examples To modify template 1 to use LCP Echo for link quality management, use the command:

```
SET PPP TEMPLATE=1 ECHO=ON
```

See Also CREATE PPP TEMPLATE
DESTROY PPP TEMPLATE
DISABLE PPP TEMPLATE DEBUG
ENABLE PPP TEMPLATE DEBUG
RESET PPP
SHOW PPP TEMPLATE
SHOW PPP LIMITS

SHOW PPP

Syntax SHOW PPP [=ppp-interface]

where:

■ *ppp-interface* is the PPP interface number.

Description This command displays a list of each PPP interface, users of the interface, physical interfaces that the interface is running over and the current state of the interface (Figure 3-8 on page 3-59, Table 3-5 on page 3-59).

Figure 3-8: Example output from the SHOW PPP command.

Name	Enabled	ifIndex	Over	CP	State
ppp0	YES	04		IPCP	OPENED
			isdn-remote	LCP	OPENED
			isdn-demand	LCP	OPENED

Table 3-5: Parameters displayed in the output of the SHOW PPP command.

Parameter	Meaning
Name	The name of the PPP interface.
Enabled	YES if the PPP interface is enabled; NO if it is disabled.
IfIndex	The value of ifIndex for the PPP interface.
Over	The physical layer(s) used by the PPP interface; one of <i>ISDN-callname</i> , <i>MIOXn-circuitname</i> , <i>TDM-groupname</i> or <i>ETHn-servicename</i> .
CP	A list of the network and link control protocols running over the PPP interface; one or more of "IPCP", "BCP", "LCP", "CCP", "ILCCP", "BACP" or "MULTI".
State	The state of the PPP links; one of "INITIAL", "STARTING", "CLOSED", "STOPPED", "CLOSING", "STOPPING", "REQ SENT", "ACK RCVD", "ACK SENT" or "OPENED".

Examples To display information about PPP interface 2, use the command:

```
SHOW PPP=2
```

See Also SHOW PPP CONFIG
SHOW PPP COUNT

SHOW PPP CONFIG

Syntax SHOW PPP [=ppp-interface] CONFIG

where:

■ *ppp-interface* is the PPP interface number.

Description This command displays the configuration of a PPP interface (Figure 3-9 on page 3-60, Table 3-6 on page 3-61).

Figure 3-9: Example output from the SHOW PPP CONFIG command.

Interface - description			
Parameter	Configured	Negotiated	

ppp0 - Link to Southern Regional Office		Local	Peer
Bandwidth Allocation Protocol	ON		
Bandwidth Allocation Call Mode	CALL		
Multilink Fragmentation	OFF		
Acceptable Fragment Overhead (%)	5		
Null Fragment Timer (seconds)	3		
Session Timer (seconds)	OFF		
Idle Timer (seconds)	60		
Maximum Receive Unit (bytes)	1656	NONE	NONE
Compression	ON	ON	ON
Username	NOT SET		
Password	NOT SET		
Bundle Endpoint Discr Class	0		
Bundle Endpoint Discr Value	[]		
Bundle Username	NOT SET		
isdn-btb			
Type	primary		
Restart Timer (seconds)	3		
Max-Configure	continuous		
Max-Terminate	2		
Echo Request Timer (seconds)	OFF		
Callback Mode	OFF		
Link Compression	ON	ON	ON
LQR Timer (seconds)	60	OFF	OFF
Magic Number	ON	OFF	OFF
Link Discriminator	0000	OFF	OFF
Link Endpoint Discr Class	0		
Link Endpoint Discr Value			
Authentication	PAP	NONE	NONE
Authentication Mode	INOUT		
Utilisation (%)	0		
Compression			
Algorithm	STACLZS	STACLZS	STACLZS
Stac LZS Checkmode	SEQUENCE	SEQUENCE	SEQUENCE
IP			
IP Compression Protocol	NONE	NONE	NONE
IP Pool	Test		
IP Address Request	OFF		
IP Address	192.168.1.1	192.168.1.1	192.168.1.2
Primary DNS Address	192.168.2.3	NONE	NONE
Secondary DNS Address	192.168.5.1		NONE
Primary WinS Address	192.168.5.5		NONE
Secondary WinS Address	NOT SET		NONE
Debug			
Maximum packet bytes to display	22		

Table 3-6: Parameters displayed in the output of the SHOW PPP CONFIG command.

Parameter	Meaning
Configured	This column specifies the value that has been configured for a parameter. The value may be modified by the negotiation process between the local and remote ends of the PPP link.
Negotiated/Local	For a link that is in the OPENED state, this column specifies the value that the local end of the link will use for a parameter, as a result of the negotiation process. For a link that is not in the OPENED state, this column displays the initial value for a parameter.
Negotiated/Peer	For a link that is in the OPENED state, this column specifies the value that the remote end of the link will use for a parameter, as a result of the negotiation process. For a link that is not in the OPENED state, this column displays the initial value for a parameter.
ppp<n> - <description>	The name and description of the interface. Following fields display information about the interface as a whole.
Bandwidth Allocation Protocol	Whether or not the Bandwidth Allocation Protocol is enabled on the interface; one of "ON" or "OFF".
Bandwidth Allocation Call Mode	The call mode for the Bandwidth Allocation Protocol, if the Bandwidth Allocation Protocol is enabled on the interface; one of "CALL" or "CALLBACK".
Multilink fragmentation	Whether or not multilink packets may be fragmented; one of "ON" or "OFF".
Acceptable Fragment Overhead (%)	The maximum amount of overhead allowed to be added to each packet due to variable fragmentation. If this level is exceeded when fragmentation of a packet is done using the variable fragmentation scheme, then the fixed fragmentation scheme is used instead.
Null Fragment Timer (seconds)	The time, in seconds, that the link must be idle for before a Null fragment is sent on a link in a multilink bundle.
Session Timer (seconds)	The time, in seconds, before a link is disconnected, or "OFF" if the session timer is disabled.
Idle Timer (seconds)	The length of time, in seconds, a link must be idle before it is disconnected, or "OFF" if the idle timer is disabled.
Maximum Receive Unit (bytes)	The maximum allowable length for packets received at the PPP layer. The MRU of the peer is used as the MTU of the upper layers so that they don't transmit anything that is too long for the peer to handle.
Compression	Whether or not compression is enabled for the entire PPP interface; one of "ON" or "OFF".
Username	The username used by the PPP interface for both PAP and CHAP authentication, or "NOT SET" if a username has not been set.
Password	Whether or not a password has been set for the entire PPP interface; one of "SET" or "NOT SET".
Up Rate (%utilisation)	The utilisation level on the link at which an additional channel is opened, if the interface has on-demand links.
Up Time (seconds)	The time, in seconds, that the utilisation level on the link must exceed <i>Up Rate</i> before an additional channel is opened, if the interface has on-demand links.

Table 3-6: Parameters displayed in the output of the SHOW PPP CONFIG command. (Continued)

Parameter	Meaning
Down Rate (%utilisation)	The utilisation level on the link below which a channel is closed, if the interface has on-demand links.
Down Time (seconds)	The time, in seconds, that the utilisation level on the link must be below <i>Down Rate</i> before a channel is closed, if the interface has on-demand links.
Bundle Endpoint Discr Class	The class of endpoint discriminator used to uniquely identify this link's endpoint.
Bundle Endpoint Discr Value	The value, in hexadecimal, of the endpoint discriminator used to uniquely identify this link's endpoint.
Bundle Username	The username assigned to the multilink bundle, or "NOT SET" if a username has not been set.
LCP Information	This section is repeated once for each LCP (physical interface) operating over the PPP interface.
<lcp-name>	The name of an LCP operating over this PPP interface. Following fields display information about this LCP (link).
Number of primary channels	The number of channels with a TYPE of PRIMARY carried over the ISDN call, if this physical interface is an ISDN call.
Number of secondary channels	The number of channels with a TYPE of SECONDARY carried over the ISDN call, if this physical interface is an ISDN call.
Number of demand channels	The number of channels with a TYPE of DEMAND carried over the ISDN call, if this physical interface is an ISDN call.
Type	The role of this physical interface for bandwidth on demand and leased line backup; one of "demand", "primary" or "secondary".
Restart Timer	The time, in seconds, between configure requests for this physical interface.
Max-Configure	The maximum number of configure requests sent before PPP gives up trying to open this link, or "continuous".
Max-Terminate	The maximum number of Terminate requests sent before PPP gives up trying to open this link and declares this link down, or "continuous".
Echo Request Timer (seconds)	The time, in seconds, between transmissions of LCP <i>Echo Request</i> messages when LCP <i>Echo Request/Echo Reply</i> messages are used to monitor link state, or "OFF" if LQR is used to determine link status.
Callback Mode	Whether this link will request callback, accept callback or do neither; one of "REQUEST", "ACCEPT" or "OFF".
Callback Operation	The callback operation to include in the callback request when the callback mode is REQUEST; one of "USERAUTH" or "E164NUMBER". This field is only displayed if <i>Callback Mode</i> is set to "REQUEST".
Callback Number	The callback number included in callback requests when the callback mode is REQUEST and the callback operation is E164NUMBER. This field is only displayed if <i>Callback Mode</i> is set to "REQUEST" and <i>Callback Operation</i> is set to "E164NUMBER".

Table 3-6: Parameters displayed in the output of the SHOW PPP CONFIG command. (Continued)

Parameter	Meaning
Callback Delay (tenths of a second)	The delay, in tenths of a second, between deactivating a call for callback and making the return call. This field is only displayed if <i>Callback Mode</i> is set to "ACCEPT".
Link Compression	Whether or not compression is enabled for this link rather than the entire PPP interface; one of "ON" or "OFF".
LQR Timer (seconds)	The time in seconds between LQR packets transmitted over this physical interface.
Magic Number	Whether or not the magic number option is enabled for this physical interface; one of "ON" or "OFF".
Link Discriminator	The link discriminator value for this physical interface, or "OFF" if the link discriminator LCP option is not enabled.
Link Endpoint Discr Class	The class of link endpoint discriminator assigned to this end of the physical interface.
Link Endpoint Discr Value	The value the of link endpoint discriminator assigned to this end of the physical interface, expressed in hexadecimal.
Authentication	The authentication protocol in use on this physical interface; one of "NONE", "PAP", "CHAP" or "EITHER".
Authentication Mode	Whether authentication will be requested on incoming ISDN calls, outgoing ISDN calls, or both incoming and outgoing ISDN calls; one of "IN", "OUT" or "INOUT".
Utilisation (%)	The bandwidth utilisation, as a percentage of time the interface is transmitting data, for this physical interface.
Link Compression	Information about link compression on this physical interface if link compression is enabled on this physical interface.
Algorithm	The compression algorithm used to compress packets on this physical interface; "STAC_LZS".
Stac LZS Checkmode	The check mode used by the Stac LZS compression algorithm to determine if a decompression history is unsynchronised on this physical interface; one of "NONE", "LCB", "CRC", "SEQUENCE" or "EXTENDED".
Channel Information	This section is displayed only if the LCP (physical interface) is an ISDN interface, and is repeated once for each channel in the physical interface. Basic Rate ISDN interfaces have 2 channels. Primary Rate ISDN interfaces has 30 channels.
bri<n> - channel <n> pri<n> - channel <n>	The interface and channel number of physical interfaces that are ISDN calls. Following fields display information specific to this channel.
Type	The role of this channel for bandwidth on demand and leased line backup; one of "demand", "primary" or "secondary".
Utilisation (%)	The bandwidth utilisation, as a percentage of time the interface is transmitting data, for the physical interface.
Link Compression	Whether or not compression is enabled for the link rather than the entire PPP interface; one of "ON" or "OFF".
LQR Timer (seconds)	The time in seconds between LQR packets transmitted over the physical interface.

Table 3-6: Parameters displayed in the output of the SHOW PPP CONFIG command. (Continued)

Parameter	Meaning
Magic Number	Whether or not the magic number option is enabled for the physical interface; one of "ON" or "OFF".
Link Discriminator	The link discriminator value for the physical interface, or "OFF" if the link discriminator LCP option is not enabled.
Link Endpoint Discr Class	The class of link endpoint discriminator assigned to this end of the physical interface.
Link Endpoint Discr Value	The value the of link endpoint discriminator assigned to this end of the physical interface, expressed in hexadecimal.
Authentication	The authentication protocol in use on the physical interface; one of "NONE", "PAP", "CHAP" or "EITHER".
NCP Information	This section is repeated once for each NCP configured on the PPP interface.
Link Compression	Information about link compression on the PPP interface if link compression is enabled on the PPP interface.
Algorithm	The compression algorithm to use for compressing packets on the PPP interface; "STAC_LZS".
Stac LZS Checkmode	The check mode used by the Stac LZS compression algorithm to determine if a decompression history is unsynchronised on the PPP interface; one of "NONE", "LCB", "CRC", "SEQUENCE" or "EXTENDED".
IP	Information about the IP NCP on the PPP interface, if IP is enabled on this PPP interface.
IP Compression Protocol	The IP compression protocol enabled on the PPP interface; one of "VJC" or "NONE".
IP Pool	The name of the IP address pool used to assign IP addresses for this PPP interface, or "NOT SET" if an IP address pool has not been assigned.
IP Address Request	Whether or not an IP address will be requested from the peer during IPCP negotiation; one of "ON" or "OFF".
IP Address	The IP address configured at each end of the link, "0.0.0.0" if the PPP interface is an unnumbered interface, or "NONE" if an IP address has not been assigned.
Primary DNS Address	The IP address of the primary DNS server, passed to a peer in response to an IPCP primary DNS request.
Secondary DNS Address	The IP address of the secondary DNS server, passed to a peer in response to an IPCP secondary DNS request.
Primary WinS Address	The IP address of the primary WINS server, passed to a peer in response to an IPCP primary WINS server request.
Secondary WinS Address	The IP address of the secondary WINS server, passed to a peer in response to an IPCP secondary WINS server request.
Debug	Information about debugging on the PPP interface.
Maximum packet bytes to display	The maximum number of bytes of each PPP packet displayed by the PACKET debugging option.
PPPoE Information	This section is repeated for every PPP interface operating over an Ethernet service.

Table 3-6: Parameters displayed in the output of the SHOW PPP CONFIG command. (Continued)

Parameter	Meaning
Session ID	The value, in hexadecimal, of the session ID number for the current PPP over Ethernet session. The number is allocated by the Access Concentrator to which the router is currently connected via the PPP over Ethernet session.
MAC Address of AC	The MAC address of the Access Concentrator to which the router is currently connected via the PPP over Ethernet session.

Examples To display the configuration for PPP interface 2, use the command:

```
SHOW PPP=2 CONFIG
```

See Also SHOW PPP
SHOW PPP COUNT

SHOW PPP COUNT

Syntax `SHOW PPP [=ppp-interface] COUNT [= { INTERFACE | LCP | MULTILINK | NCP }]`

where:

- *ppp-interface* is the PPP interface number.

Description This command displays counters for the interface. If INTERFACE is specified, counters from the Interfaces MIB are displayed (Figure 3-10 on page 3-65, Table 3-7 on page 3-66). If LCP is specified, counters for LCP, LQR, CCP and authentication protocols are displayed (Figure 3-11 on page 3-67, Table 3-8 on page 3-68). If MULTILINK is specified, counters for the multilink protocol are displayed (Figure 3-12 on page 3-71, Table 3-9 on page 3-71). If NCP is specified, counters for the NCPs are displayed (Figure 3-13 on page 3-72, Table 3-10 on page 3-72). If a category is not specified all counters are displayed, including those for BAP and BACP (Table 3-11 on page 3-73).

Figure 3-10: Example output from the SHOW PPP COUNT=INTERFACE command.

ppp0	1519 seconds	Last change at:	974 seconds
Interface Counters			
ifInOctets	116554	ifOutOctets	91792
ifInUcastPkts	0	ifOutUcastPkts	0
ifInNUcastPkts	2098	ifOutNUcastPkts	1538
ifInDiscards	0	ifOutDiscards	0
ifInErrors	3	ifOutErrors	0
ifInUnknownProtos	0	ifOutQLen	0

Table 3-7: Parameters displayed in the output of the SHOW PPP COUNT=INTERFACE command.

Parameter	Meaning
ppp0	The interface name.
seconds	The time (in seconds) since the interface was last re-initialised.
Last change at	The time (in seconds) since the interface entered its current operational state.
ifInOctets	The total number of octets received over the interface, including two octets per frame for PPP address and control information, two octets per frame for the FCS, one octet per frame for a flag and two octets per frame for the PPP header (six for multilink), and the number of octets in the user data packets and PPP control packets.
ifInUcastPkts	The total number of subnetwork-unicast packets delivered to a higher-layer protocol.
ifInNUcastPkts	The total number of non-unicast packets delivered to a higher-layer protocol.
ifInDiscards	The total number of inbound packets which were chosen to be discarded even though no errors had been detected to prevent their being deliverable to a higher-layer protocol. One reason for discarding such packets would be to free up buffer space.
ifInErrors	The total number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol.
ifInUnknownProtos	The total number of packets received via the interface which were discarded because of an unknown or unsupported protocol.
ifOutOctets	The total number of octets transmitted over the interface, including two octets per frame for PPP address and control information, two octets per frame for the FCS, one octet per frame for a flag and two octets per frame for the PPP header (six for multilink), and the number of octets in the user data packets and PPP control packets.
ifOutUcastPkts	The total number of packets that higher-layer protocols requested be transmitted to a subnetwork-unicast address, including those that were discarded or not sent.
ifOutNUcastPkts	The total number of packets that higher-layer protocols requested be transmitted to a non-unicast address, including those that were discarded or not sent.
ifOutDiscards	The total number of outbound packets which were chosen to be discarded even though no errors had been detected to prevent their being transmitted. One possible reason for discarding such a packet could be to free up buffer space.
ifOutErrors	The total number of outbound packets that contained errors preventing them from being transmitted.
ifOutQLen	The length of the output packet queue.

Figure 3-11: Example output from the SHOW PPP COUNT=LCP command.

CCP			
inOctets	52456	outOctets	38959
inUserPkts	2101	outUserPkts	1538
inConfigureRequest	3	outConfigureRequest	3
inConfigureAcknowledge	3	outConfigureAcknowledge	3
inConfigureNAK	0	outConfigureNAK	0
inConfigureReject	0	outConfigureReject	0
inTerminateRequest	0	outTerminateRequest	0
inTerminateAcknowledge	0	outTerminateAcknowledge	0
inCodeReject	0	outCodeReject	0
decodeSuccesses	2098	encodeSuccesses	1538
decodeFailures	3	encodeFailures	0
decodeDiscards	0	encodeDiscards	0
inResetRequests	8	outResetRequests	3
inResetAcks	3	outResetAcks	2
encoEventsWithLcpDown	0		
LQM OVER: isdn-remote			
lqrFailures	0	loopbacksDetected	0
inLQRs	16	outLQRs	16
inPktLost	0	outPktLost	0
inOctetLost	0	outOctetLost	0
		outLQRsLost	0
		outLQRsTransit	0
PAP OVER: isdn-remote			
inRequest	1	outRequest	0
inAck	0	outAck	1
inNak	0	outNak	0
LCP OVER: isdn-remote			
inOctets	25316	outOctets	19158
inUserPkts	929	outUserPkts	749
inConfigureRequest	3	outConfigureRequest	7
inConfigureAcknowledge	3	outConfigureAcknowledge	3
inConfigureNAK	0	outConfigureNAK	0
inConfigureReject	3	outConfigureReject	0
inTerminateRequest	0	outTerminateRequest	1
inTerminateAcknowledge	1	outTerminateAcknowledge	0
inCodeReject	0	outCodeReject	0
inProtocolReject	3	outProtocolReject	0
inEchoRequest	0	outEchoRequest	0
inEchoReply	0	outEchoReply	0
inDiscardRequest	0	outDiscardRequest	0
echoFailures	0	badEchoReplies	0

Table 3-8: Parameters displayed in the output of the SHOW PPP COUNT=LCP command.

Parameter	Meaning
CCP ILCCP OVER: <interface>	Information about the compression control protocol (CCP) or ILCCP and the physical interface over which ILCCP is running.
inOctets	The number of octets received by the compression protocol. This includes two octets per frame for the PPP compression header, the number of octets of compressed data received, and the number of octets in control protocol packets (CCP). For multilinks an extra six octets per frame are included for the multilink header.
inUserPkts	The number of packets received by the compression control protocol.
inConfigureRequest	The number of <i>Configure-Request</i> packets received by the compression control protocol.
inConfigureAcknowledge	The number of <i>Configure-Acknowledge</i> packets received by the compression control protocol.
inConfigureNAK	The number of <i>Configure-NAK</i> packets received by the compression control protocol.
inConfigureReject	The number of <i>Configure-Reject</i> packets received by the compression control protocol.
inTerminateRequest	The number of <i>Terminate-Request</i> packets received by the compression control protocol.
inTerminateAcknowledge	The number of <i>Terminate-Acknowledge</i> packets received by the compression control protocol.
inCodeReject	The number of <i>Code-Reject</i> packets received by the compression control protocol.
decodeSuccesses	The number of packets successfully decoded by the compression control protocol.
decodeFailures	The number of packets that failed to be decoded correctly by the compression control protocol.
decodeDiscards	The number of packets that were discarded by the compression control protocol.
inResetRequests	The number of <i>Reset-Request</i> packets received to reset the compression history.
inResetACKs	The number of <i>Reset-Acknowledge</i> packets received to reset the compression history.
encoEventsWithLcpDown	The number of times the PPP interface received an event from the ENCO module when the interface's LCP was not in the OPENED state.
outOctets	The number of octets transmitted by the compression protocol. This includes two octets per frame for the PPP compression header, the number of octets of compressed data transmitted, and the number of octets in control protocol packets (CCP). For multilinks an extra six octets per frame are included for the multilink header.
outUserPkts	The number of packets transmitted by the compression control protocol.
outConfigureRequest	The number of <i>Configure-Request</i> packets transmitted by the compression control protocol.

Table 3-8: Parameters displayed in the output of the SHOW PPP COUNT=LCP command. (Continued)

Parameter	Meaning
outConfigureAcknowledge	The number of <i>Configure-Acknowledge</i> packets transmitted by the compression control protocol.
outConfigureNAK	The number of <i>Configure-NAK</i> packets transmitted by the compression control protocol.
outConfigureReject	The number of <i>Configure-Reject</i> packets transmitted by the compression control protocol.
outTerminateRequest	The number of <i>Terminate-Request</i> packets transmitted by the compression control protocol.
outTerminateAcknowledge	The number of <i>Terminate-Acknowledge</i> packets transmitted by the compression control protocol.
outCodeReject	The number of <i>Code-Reject</i> packets transmitted by the compression control protocol.
encodeSuccesses	The number of packets successfully encoded.
encodeFailures	The number of packets that failed to be encoded correctly.
encodeDiscards	The number of packets to be encoded that were discarded.
outResetRequests	The number of <i>Reset-Request</i> packets transmitted to reset the compression history.
outResetACKs	The number of <i>Reset-Acknowledge</i> packets transmitted to reset the compression history.
LQM OVER: <interface>	Information about LQR and the physical interface over which LQR is running.
lqrFailures	The number of times the LQR timer has timed out.
loopbacksDetected	The number of times the link entered loopback mode.
inLQRs	The number of LQR packets received.
inPktLost	The number of inbound LQR packets lost.
inOctetLost	The number of inbound LQR octets lost.
outLQRs	The number of LQR packets transmitted.
outPktLost	The number of outbound LQR packets lost.
outOctetLost	The number of outbound LQR octets lost.
outLQRsLost	The number of outbound LQR packets lost.
outLQRsTransit	The number of outbound LQR packets in transit.
PAP OVER: <interface>	Information about PAP and the physical interface over which PAP is running.
inRequest	The number of PAP <i>Authenticate-Request</i> packets received.
inAck	The number of PAP <i>Authenticate-Acknowledgement</i> packets received.
inNak	The number of PAP <i>Authenticate-Negative-Acknowledgement</i> packets received.
outRequest	The number of PAP <i>Authenticate-Request</i> packets transmitted.
outAck	The number of PAP <i>Authenticate-Acknowledgement</i> packets transmitted.
outNak	The number of PAP <i>Authenticate-Negative-Acknowledgement</i> packets transmitted.

Table 3-8: Parameters displayed in the output of the SHOW PPP COUNT=LCP command. (Continued)

Parameter	Meaning
CHAP OVER: <interface>	Information about CHAP and the physical interface over which CHAP is running.
inChallenge	The number of CHAP <i>Challenge</i> packets received for.
inResponse	The number of CHAP <i>Response</i> packets received.
inSuccess	The number of CHAP <i>Success</i> packets received.
inFailure	The number of CHAP <i>Failure</i> packets received.
outChallenge	The number of CHAP <i>Challenge</i> packets transmitted.
outResponse	The number of CHAP <i>Response</i> packets transmitted.
outSuccess	The number of CHAP <i>Success</i> packets transmitted.
outFailure	The number of CHAP <i>Failure</i> packets transmitted.
LCP OVER: <interface>	Information about the LCP and the physical interface over which LCP is running.
inOctets	The number of octets received by the link control protocol. This includes the number of octets in control protocol packets (e.g. LCP, LQR, PAP, CHAP), plus the number of octets of data received. The number of octets of data will be equal to the number of inOctets recorded for compression if enabled. If compression is not enabled the number of inOctets of data will be equal to the sum of the inOctets recorded for all the user protocols on this link. For multilinks an extra six octets per frame are included for the multilink header.
inUserPkts	The number of packets received for the LCP.
inConfigureRequest	The number of <i>Configure-Request</i> packets received for the LCP.
inConfigureAcknowledge	The number of <i>Configure-Acknowledge</i> packets received for the LCP.
inConfigureNAK	The number of <i>Configure-NAK</i> packets received for the LCP.
inConfigureReject	The number of <i>Configure-Reject</i> packets received for the LCP.
inTerminateRequest	The number of <i>Terminate-Request</i> packets received for the LCP.
inTerminateAcknowledge	The number of <i>Terminate-Acknowledge</i> packets received for the LCP.
inCodeReject	The number of <i>Code-Reject</i> packets received for the LCP.
inProtocolReject	The number of <i>Protocol Reject</i> packets received for the LCP.
inEchoRequest	The number of <i>Echo Request</i> packets received for the LCP.
inEchoReply	The number of <i>Echo Reply</i> packets received for the LCP.
inDiscardRequest	The number of <i>Discard Request</i> packets received for the LCP.
echoFailures	The number of times the ECHO timer has timed out.

Table 3-8: Parameters displayed in the output of the SHOW PPP COUNT=LCP command. (Continued)

Parameter	Meaning
outOctets	The number of octets transmitted by the LCP. This includes the number of octets in control protocol packets (e.g. LCP, LQR, PAP, CHAP), plus the number of octets of data transmitted. The number of octets of data will be equal to the number of outOctets recorded for compression if enabled. If compression is not enabled the number of outOctets of data will be equal to the sum of the outOctets recorded for all the user protocols on this link. For multilinks an extra six octets per frame are included for the multilink header.
outUserPkts	The number of packets sent for the LCP.
outConfigureRequest	The number of <i>Configure-Request</i> packets sent for the LCP.
outConfigureAcknowledge	The number of <i>Configure-Acknowledge</i> packets sent for the LCP.
outConfigureNAK	The number of <i>Configure-NAK</i> packets sent for the LCP.
outConfigureReject	The number of <i>Configure-Reject</i> packets sent for the LCP.
outTerminateRequest	The number of <i>Terminate-Request</i> packets sent for the LCP.
outTerminateAcknowledge	The number of <i>Terminate-Acknowledge</i> packets sent for the LCP.
outCodeReject	The number of <i>Code-Reject</i> packets sent for the LCP.
outProtocolReject	The number of <i>Protocol Reject</i> packets sent for the LCP.
outEchoRequest	The number of <i>Echo Request</i> packets sent for the LCP.
outEchoReply	The number of <i>Echo Reply</i> packets sent for the LCP.
outDiscardRequest	The number of <i>Discard Request</i> packets sent for the LCP.
badEchoReplies	The number of <i>Echo Reply</i> packets received with a different ID than the original <i>Echo Request</i> packet.

Figure 3-12: Example output from the SHOW PPP COUNT=MULTILINK command.

Multilink Counters			
inWholeFragments	1538	outWholeFragments	1538
inStartFragments	0	outStartFragments	0
inMiddleFragments	0	outMiddleFragments	0
inEndFragments	0	outEndFragments	0
inNullFragments	54	outNullFragments	54

Table 3-9: Parameters displayed in the output of the SHOW PPP COUNT=MULTILINK command.

Parameter	Meaning
inWholeFragments	The number of multilink encapsulated fragments received that contain a whole packet.
inStartFragments	The number of multilink encapsulated fragments received that contain the start of a packet.
inMiddleFragments	The number of multilink encapsulated fragments received that contain part of a packet that is not the start or the end.

Table 3-9: Parameters displayed in the output of the SHOW PPP COUNT=MULTILINK command. (Continued)

Parameter	Meaning
inEndFragments	The number of multilink encapsulated fragments received that contain the end of a packet.
inNullFragments	The number of NULL multilink encapsulated fragments that have been received.
outWholeFragments	The number of multilink encapsulated fragments transmitted that contain a whole packet.
outStartFragments	The number of multilink encapsulated fragments transmitted that contain the start of a packet.
outMiddleFragments	The number of multilink encapsulated fragments transmitted that contain part of a packet that is not the start or the end.
outEndFragments	The number of multilink encapsulated fragments transmitted that contain the end of a packet.
outNullFragments	The number of NULL multilink encapsulated fragments that have been transmitted.

Figure 3-13: Example output from the SHOW PPP COUNT=NCP command.

IPCP			
inOctets	63611	outOctets	91768
inUserPkts	2098	outUserPkts	1538
inConfigureRequest	1	outConfigureRequest	1
inConfigureAcknowledge	1	outConfigureAcknowledge	1
inConfigureNAK	0	outConfigureNAK	0
inConfigureReject	0	outConfigureReject	0
inTerminateRequest	0	outTerminateRequest	0
inTerminateAcknowledge	0	outTerminateAcknowledge	0
inCodeReject	0	outCodeReject	0

Table 3-10: Parameters displayed in the output of the SHOW PPP COUNT=NCP command.

Parameter	Meaning
inOctets	The number of octets received by the network protocol. This includes two octets per frame for the PPP protocol header, the number of octets of user data to be passed up to the user protocol, and the number of octets in control protocol packets (e.g. IPCP, ATCP).
inUserPkts	The number of packets received for the network control protocol.
inConfigureRequest	The number of Configure-Request packets received for the network control protocol.
inConfigureAcknowledge	The number of Configure-Acknowledge packets received for the network control protocol.
inConfigureNAK	The number of Configure-NAK packets received for the network control protocol.
inConfigureReject	The number of Configure-Reject packets received for the network control protocol.

Table 3-10: Parameters displayed in the output of the SHOW PPP COUNT=NCP command. (Continued)

Parameter	Meaning
inTerminateRequest	The number of Terminate-Request packets received for the network control protocol.
inTerminateAcknowledge	The number of Terminate-Acknowledge packets received for the network control protocol.
inCodeReject	The number of Code-Reject packets received for the network control protocol.
outOctets	The number of octets transmitted by the network protocol. This includes two octets per frame for the PPP protocol header, the number of octets of user data passed down from the user protocol, and the number of octets in control protocol packets (e.g. IPCP, ATCP).
outUserPkts	The number of packets sent for the network control protocol.
outConfigureRequest	The number of Configure-Request packets sent for the network control protocol.
outConfigureAcknowledge	The number of Configure-Acknowledge packets sent for the network control protocol.
outConfigureNAK	The number of Configure-NAK packets sent for the network control protocol.
outConfigureReject	The number of Configure-Reject packets sent for the network control protocol.
outTerminateRequest	The number of Terminate-Request packets sent for the network control protocol.
outTerminateAcknowledge	The number of Terminate-Acknowledge packets sent for the network control protocol.
outCodeReject	The number of Code-Reject packets sent for the network control protocol.

Table 3-11: Parameters displayed in the output of the SHOW PPP COUNT command for BAP and BACP.

Parameter	Meaning
BAP	Information about the operation of BAP.
inCallReq	The number of <i>Call-Request</i> packets received by the BAP protocol.
inCallResp	The number of <i>Call-Response</i> packets received by the BAP protocol.
inCallbackReq	The number of <i>Callback-Request</i> packets received by the BAP protocol.
inCallbackResp	The number of <i>Callback-Response</i> packets received by the BAP protocol.
inLinkDropQueryReq	The number of <i>Link-Drop-Query-Request</i> packets received by the BAP protocol.
inLinkDropQueryResp	The number of <i>Link-Drop-Query-Response</i> packets received by the BAP protocol.
inCallStatusInd	The number of <i>Call-Status-Indication</i> packets received by the BAP protocol.

Table 3-11: Parameters displayed in the output of the SHOW PPP COUNT command for BAP and BACP. (Continued)

Parameter	Meaning
inCallStatusResp	The number of <i>Call-Status-Response</i> packets received by the BAP protocol.
inErrors	The number of packets received by the BAP protocol which contained errors.
inDiscards	The number of packets received by the BAP protocol that were discarded.
outCallReq	The number of <i>Call-Request</i> packets transmitted by the BAP protocol.
outCallResp	The number of <i>Call-Response</i> packets transmitted by the BAP protocol.
outCallbackReq	The number of <i>Callback-Request</i> packets transmitted by the BAP protocol.
outCallbackResp	The number of <i>Callback-Response</i> packets transmitted by the BAP protocol.
outLinkDropQueryReq	The number of <i>Link-Drop-Query-Request</i> packets transmitted by the BAP protocol.
outLinkDropQueryResp	The number of <i>Link-Drop-Query-Response</i> packets transmitted by the BAP protocol.
outCallStatusInd	The number of <i>Call-Status-Indication</i> packets transmitted by the BAP protocol.
outCallStatusResp	The number of <i>Call-Status-Response</i> packets transmitted by the BAP protocol.
BACP	Information about the operation of BACP.
inOctets	The number of octets received by the BACP protocol.
inUserPkts	The number of packets received by the BACP protocol.
inConfigureRequest	The number of <i>Configure-Request</i> packets received by the BACP protocol.
inConfigureAcknowledge	The number of <i>Configure-Acknowledge</i> packets received by the BACP protocol.
inConfigureNAK	The number of <i>Configure-NAK</i> packets received by the BACP protocol.
inConfigureReject	The number of <i>Configure-Reject</i> packets received by the BACP protocol.
inTerminateRequest	The number of <i>Terminate-Request</i> packets received by the BACP protocol.
inTerminateAcknowledge	The number of <i>Terminate-Acknowledge</i> packets received by the BACP protocol.
inCodeReject	The number of <i>Code-Reject</i> packets received by the BACP protocol.
outOctets	The number of octets transmitted by the BACP protocol.
outUserPkts	The number of packets transmitted by the BACP protocol.
outConfigureRequest	The number of <i>Configure-Request</i> packets transmitted by the BACP protocol.
outConfigureAcknowledge	The number of <i>Configure-Acknowledge</i> packets transmitted by the BACP protocol.

Table 3-11: Parameters displayed in the output of the SHOW PPP COUNT command for BAP and BACP. (Continued)

Parameter	Meaning
outConfigureNAK	The number of <i>Configure-NAK</i> packets transmitted by the BACP protocol.
outConfigureReject	The number of <i>Configure-Reject</i> packets transmitted by the BACP protocol.
outTerminateRequest	The number of <i>Terminate-Request</i> packets transmitted by the BACP protocol.
outTerminateAcknowledge	The number of <i>Terminate-Acknowledge</i> packets transmitted by the BACP protocol.
outCodeReject	The number of <i>Code-Reject</i> packets transmitted by the BACP protocol.

Examples To display the interface counters for PPP interface 1, use the command:

```
SHOW PPP=1 COUNT=INTERFACE
```

See Also SHOW PPP
SHOW PPP CONFIG
SHOW PPP IDLETIMER
SHOW PPP MULTILINK

SHOW PPP DEBUG

Syntax SHOW PPP [=ppp-interface] DEBUG

where:

- *ppp-interface* is the PPP interface number.

Description This command displays the debugging options that are currently enabled for the specified or all PPP interfaces (Figure 3-14 on page 3-75, Table 3-12 on page 3-76).

Figure 3-14: Example output from the SHOW PPP DEBUG command.

Interface	Enabled Debug Modes
ppp0	AUTH, LCP, PKT, UTILISATION

Table 3-12: Parameters displayed in the output of the SHOW PPP DEBUG command.

Parameter	Meaning
Interface	The interface name.
Enabled Debug Modes	The list of currently enabled debug modes for the interface; one or more of "AUTH", "BAPPKT", "BAPSTATE", "CALLBACK", "DEMAND", "ENCO", "LCP", "NCP", "PKT" or "UTILISATION".

Examples To display the debugging options set for all PPP interfaces, use the command:

```
SHOW PPP DEBUG
```

See Also DISABLE PPP DEBUG
ENABLE PPP DEBUG

SHOW PPP IDLETIMER

Syntax SHOW PPP [=ppp-interface] IDLETIMER

where:

- *ppp-interface* is the PPP interface number.

Description This command displays the configured and current values of the PPP idle timer for the specified or all PPP interfaces (Figure 3-15 on page 3-76, Table 3-13 on page 3-76).

Figure 3-15: Example output from the SHOW PPP IDLETIMER command.

Interface	Configured Idle Time	Idle Timer Value
ppp0	60	EXPIRED

Table 3-13: Parameters displayed in the output of the SHOW PPP IDLETIMER command.

Parameter	Meaning
ppp0	The interface name.
Configured Idle Time	The configured value, in seconds, of the idle timer for the interface.
Idle Timer Value	The current value, in seconds, of the idle timer for the interface, or "EXPIRED" if the timer has expired.

Examples To display the idle timers for all PPP interfaces, use the command:

```
SHOW PPP IDLETIMER
```


See Also SHOW PPP
 SHOW PPP CONFIG
 SHOW PPP COUNT
 SHOW PPP MULTILINK

SHOW PPP LIMITS

Syntax SHOW PPP [=ppp-interface] LIMITS

where:

■ *ppp-interface* is the PPP interface number.

Description This command displays information about the accumulated up-time and input/output data counters for the specified PPP interface or all PPP interfaces. It shows the current values of the counters, and the threshold limits if they are defined.

Figure 3-16: Example output from the SHOW PPP Limits command.

Name		Current	Limit	Remaining

ppp0	Up Time	16:12	25 hrs	08:47
	In Data	EXCEEDED	50 MB	0.0 MB
	Out Data	21.5 MB	Unlimited	--
	Total Data	71.5 MB	Unlimited	--

Table 3-14: Parameters displayed the SHOW PPP LIMITS command output

Parameter	Meaning
Name	The name of the PPP interface.
Over	The physical layer(s) used by the PPP interface; one of ISDN-callname, MIOXn-circuitname, TDM-groupname or ETHn-servicename.
Up-Time	The cumulative up-time, in hours, for the interface.
In Data	The cumulative input data throughput, in megabytes, for the interface.
Out Data	The cumulative output data throughput, in megabytes, for the interface.
Total Data	The cumulative total data throughput, in megabytes, for the interface.
Current	The current value of the cumulative counter, or "EXCEEDED" if the interface counter has exceeded the corresponding threshold limit.
Limit	The threshold limits for the interface, or "Unlimited" if no value has been specified.
Remaining	The remaining time/data throughput allowed before the limit is exceeded and the link closed.

Example To display the up-time and input/output data counters for PPP interface 0, use the command:

```
SHOW PPP=0 LIMITS
```

See also CREATE PPP
CREATE PPP TEMPLATE
RESET PPP
SET PPP
SET PPP TEMPLATE

SHOW PPP MULTILINK

Syntax SHOW PPP [=ppp-interface] MULTILINK

where:

■ *ppp-interface* is the PPP interface number.

Description This command displays information about the multilink bundle for the specified or all PPP interfaces (Figure 3-17 on page 3-78, Table 3-15 on page 3-79).

Figure 3-17: Example output from the SHOW PPP MULTILINK command.

Interface		Value
Parameter		

ppp0		
Multilink Enabled		Yes
Fragmentation Enabled		No
Acceptable fragmentation overhead for VF scheme (%)		5
Minimum packet size for fragmentation using VF scheme (bytes)		120
Null fragment timer (seconds)		3
Number of links in bundle		4
Total bandwidth of bundle (bps)		256000
Number of packets fragmented using VF scheme		0
Number of packets fragmented using FF scheme		0
Number of packets not fragmented		971
Next output sequence number		972
Minimum sequence number received on bundle		863
Next expected sequence number		866
Length of receive queue		0
Discards from receive queue		0

Table 3-15: Parameters displayed in the output of the SHOW PPP MULTILINK command.

Parameter	Meaning
ppp0	The interface name.
Multilink Enabled	Whether or multilink is enabled on this PPP interface; one of "Yes" or "No".
Fragmentation Enabled	Whether or fragmentation is enabled on this PPP interface; one of "Yes" or "No".
Acceptable fragmentation overhead for VF scheme (%)	The maximum acceptable overhead for fragmentation using the variable fragmentation scheme, as a percentage of packet size.
Minimum packet size for fragmentation using VF scheme (bytes)	The minimum size packet that may be fragmented using the variable fragmentation scheme.
Null fragment timer (seconds)	The time, in seconds, the link must be idle before a null fragment is transmitted.
Number of links in bundle	The number of links in the multilink bundle.
Total bandwidth of bundle (bps)	The total bandwidth, in bits per second, of the multilink bundle.
Number of packets fragmented using VF scheme	The number of packets that have been fragmented using the variable fragmentation scheme.
Number of packets fragmented using FF scheme	The number of packets that have been fragmented using the fixed fragmentation scheme.
Number of packets not fragmented	The number of packets that have not been fragmented.
Next output sequence number	The sequence number to use in the next transmission over the multilink bundle.
Minimum sequence number received on bundle	The lowest sequence number received via the multilink bundle.
Next expected sequence number	The next sequence number expected via the multilink bundle.
Length of receive queue	The current length of the receive queue.
Discards from receive queue	The number of packets discarded from the receive queue due to lost packets causing sequence number synchronisation to be lost.

Examples To display multilink information for all PPP interfaces, use the command:

```
SHOW PPP MULTILINK
```

See Also SHOW PPP
SHOW PPP CONFIG
SHOW PPP COUNT
SHOW PPP IDLETIMER

SHOW PPP NAMESERVER

Syntax SHOW PPP NAMESERVER

Description This command displays information about the currently configured global DNS and WINS servers (Figure 3-18 on page 3-80, Table 3-16 on page 3-80).

Figure 3-18: Example output from the SHOW PPP NAMESERVER command.

```

Name Server                                Address
-----
Primary DNS ..... 192.168.2.3
Secondary DNS ..... 192.168.5.1
Primary WinS ..... 192.168.5.5
Secondary WinS ..... Not Set
-----

```

Table 3-16: Parameters displayed in the output of the SHOW PPP NAMESERVER command.

Parameter	Meaning
Primary DNS Address	The IP address of the primary DNS server, passed to a peer in response to an IPCP primary DNS request.
Secondary DNS Address	The IP address of the secondary DNS server, passed to a peer in response to an IPCP secondary DNS request.
Primary WinS Address	The IP address of the primary WINS server, passed to a peer in response to an IPCP primary WINS server request.
Secondary WinS Address	The IP address of the secondary WINS server, passed to a peer in response to an IPCP secondary WINS server request.

Examples To display the currently configure DNS and WINS servers, use the command:

```
SHOW PPP NAMESERVER
```

See Also SET PPP
SHOW PPP

SHOW PPP TEMPLATE

Syntax SHOW PPP TEMPLATE[=*template*] [DEBUG]

where:

- *template* is a number in the range 0 to 31.

Description This command displays information about PPP templates.

The TEMPLATE parameter specifies the number of the template to display. If a template is not specified, information about all templates, including the default template, is displayed. If a template is specified, information about the specified template is displayed (Figure 3-19 on page 3-81, Table 3-17 on

page 3-81). If no templates have been defined, the default template is displayed.

If DEBUG is specified, the debugging modes enabled for the template or all templates are displayed (Figure 3-20 on page 3-83, Table 3-18 on page 3-83).

Figure 3-19: Example output from the SHOW PPP TEMPLATE command.

Template - Description		
Parameter		Value

pppt0 - Template for ISDN calls from Head Office		
Multilink		ON
Maximum links		4
Bandwidth Allocation Protocol		ON
Bandwidth Allocation Call Mode		CALL
Multilink fragmentation		OFF
Acceptable Fragment Overhead (%)		5
Null Fragment Timer (seconds)		3
Idle Timer (seconds)		OFF
Compression		ON
Compression Algorithm		STACLZS
Compression Checkmode		LCB
Username		NOT SET
Password		NOT SET
Login Servers		USER
IP Pool		NOT SET
Request IP Address		NO
Link		
Authentication		NONE
Callback Mode		OFF
Callback Operation		USER
Callback Number		-
Callback Delay (seconds)		5
Echo Timer (seconds)		10
LQR Timer (seconds)		60
Magic Number		ON
Restart Timer (seconds)		3
Debug		
Maximum packet bytes to display		32

Table 3-17: Parameters displayed in the output of the SHOW PPP TEMPLATE command.

Parameter	Meaning
pppT<template> - <description>	The number and description of the template.
Multilink	Whether or not dynamic PPP calls can be multilinked together; one of "ON" or "OFF".
Maximum links	The maximum number of links allowed in a multilink bundle created with this template.
Bandwidth Allocation Protocol	Whether or not the Bandwidth Allocation Protocol is enabled; one of "ON" or "OFF".
Bandwidth Allocation Call Mode	The call mode for the Bandwidth Allocation Protocol, if the Bandwidth Allocation Protocol is enabled; one of "CALL" or "CALLBACK".

Table 3-17: Parameters displayed in the output of the SHOW PPP TEMPLATE command. (Continued)

Parameter	Meaning
Multilink fragmentation	Whether or not multilink packets may be fragmented; one of "ON" or "OFF".
Acceptable Fragment Overhead(%)	The maximum amount of overhead allowed to be added to each packet due to variable fragmentation. If this level is exceeded when fragmentation of a packet is done using the variable fragmentation scheme, then the fixed fragmentation scheme is used instead.
Null Fragment Timer	The time, in seconds, that the link must be idle for before a Null fragment is sent on a link in a multilink bundle.
Idle Timer (seconds)	The length of time, in seconds, a link must be idle before it is disconnected, or "OFF" if the idle timer is disabled.
Compression	Whether or not compression is enabled; one of "ON" or "OFF".
Compression Algorithm	The compression algorithm to use for compressing packets; "STACZS".
Compression Checkmode	The check mode used by the compression algorithm to determine when a decompression history becomes unsynchronised; one of "SEQUENCE", "LCB", "CRC16" or "CRCCITT".
Encryption	Whether or not encryption is enabled; one of "ON" or "OFF".
Username	The username used by the PPP interface for both PAP and CHAP authentication, or "NOT SET" if a username has not been set.
Password	Whether or not a password has been set for the entire PPP interface; one of "SET" or "NOT SET".
Login Servers	The authentication servers to use; one of "USER" or "NOT SET" if a login server has not been set.
IP Pool	The name of the IP address pool used to assign IP addresses for this PPP interface, or "NOT SET" if an IP address pool has not been assigned.
Request IP Address	Whether or not an IP address will be requested from the peer during IPCP negotiation; one of "ON" or "OFF".
Authentication	The authentication protocol in use; one of "NONE", "PAP", "CHAP" or "EITHER".
Callback Mode	Whether the link will request callback, accept callback or do neither; one of "REQUEST", "ACCEPT" or "OFF".
Callback Operation	The callback operation included in callback requests; one of "USERAUTH" or "E164NUMBER".
Callback Number	The callback number to include in callback requests when the callback operation is E164NUMBER.
Callback Delay (seconds)	The delay, in seconds, between deactivating a call for callback and making the call back to the peer.
Echo Timer (seconds)	The interval, in seconds, between transmissions of LCP <i>Echo Request</i> messages.
LQR Timer (seconds)	The time in seconds between LQR packets transmitted over the physical interface.

Table 3-17: Parameters displayed in the output of the SHOW PPP TEMPLATE command. (Continued)

Parameter	Meaning
Magic Number	Whether or not the magic number option is enabled; one of "ON" or "OFF".
Restart Timer	The time in seconds between configure requests for the physical interface.
Maximum packet bytes to display	The maximum number of bytes of each PPP packet displayed by the PACKET debugging option.

Figure 3-20: Example output from the SHOW PPP TEMPLATE DEBUG command.

Template	Call	Enabled Debug Modes
-----	-----	-----
pppT0		PKT, LCP, NCP
-----	-----	-----

Table 3-18: Parameters displayed in the output of the SHOW PPP TEMPLATE DEBUG command.

Parameter	Meaning
Template	The name of a PPP template.
Call	The lower layer call using this template, if any.
Enabled Debug Modes	The debugging modes enabled for the template (and call); one or more of "AUTH", "BAPPKT", "BAPSTATE", "CALLBACK", "DEMAND", "ENCO", "LCP", "NCP", "PKT" or "UTILISATION".

Examples To display the configuration for all templates, use the command:

```
SHOW PPP TEMPLATE
```

To display the configuration for template 1, use the command:

```
SHOW PPP TEMPLATE=1
```

To display the debugging modes enabled for template 3, use the command:

```
SHOW PPP TEMPLATE=3 DEBUG
```

See Also CREATE PPP TEMPLATE
 DESTROY PPP TEMPLATE
 DISABLE PPP TEMPLATE DEBUG
 ENABLE PPP TEMPLATE DEBUG
 SET PPP TEMPLATE

SHOW PPP TXSTATUS

Syntax SHOW PPP [=ppp-interface] TXSTATUS

where:

- *template* is a number in the range 0 to 31.

Description This command displays information about the status of a PPP transmission queue for the specified interface of all interfaces (Figure 3-21 on page 3-84, Table 3-19 on page 3-84).

Figure 3-21: Example output from the SHOW PPP TXSTATUS command.

Interface	Value
Parameter	
-----	-----
ppp0	
Interface transmission queue length	0
isdn-remote	
Packets started transmission	198
Packets being transmitted	0
Packets lost during transmission	3
Packets finished transmission	195
Packets discarded in pipe	0
Link transmission queue length	0
Driver bandwidth (bps)	48000
Driver transmission delay (ms)	0
Driver transmission status	Ready
-----	-----

Table 3-19: Parameters displayed in the output of the SHOW PPP TXSTATUS command.

Parameter	Meaning
ppp<n>	The name of a PPP interface.
Interface transmission queue length	The length of the output queue for this PPP interface.
<physical-interface>	The name of a physical interface or channel forming part of this PPP interface.
Packets started transmission	The total number of packets that higher-layer protocols requested be transmitted to a non-unicast address, including those that were discarded or not sent.
Packets being transmitted	The number of packets currently being transmitted on this physical interface or channel.
Packets lost during transmission	The number of packets lost during transmission on this physical interface or channel.
Packets finished transmission	The number of packets that have been transmitted and acknowledged on this physical interface or channel.
Packets discarded in pipe	The number of packets that were discarded on this physical interface or channel.

Table 3-19: Parameters displayed in the output of the SHOW PPP TXSTATUS command. (Continued)

Parameter	Meaning
Link transmission queue length	The length of the output queue for this physical interface or channel.
Driver bandwidth (bps)	The bandwidth of the layer 1 device driver for this physical interface or channel.
Driver transmission delay (ms)	The delay, in milliseconds, in the layer 1 device driver for this physical interface or channel.
Driver transmission status	The status of the layer 1 device driver for this physical interface or channel; one of "busy" or "ready".

Examples To display the status of the PPP transmission queue for interface ppp0, use the command:

```
SHOW PPP=0 TXSTATUS
```

See Also SET PPP
SHOW PPP

Chapter 4

Integrated Services Digital Network (ISDN)

Introduction	4-3
Basic Rate Access	4-3
Support for ISDN	4-6
BRI Physical Layer	4-7
Configuring and Controlling the Basic Rate Interface	4-7
Examining the Status of the Basic Rate Interface	4-9
Monitoring Operation of the Basic Rate Interface	4-11
LAPD	4-11
Operation	4-11
Packet mode support	4-12
Fault Finding	4-12
Default Setup	4-13
Addressing	4-13
Frame Control Fields	4-14
Non-Associated Signalling	4-15
Q.931	4-15
Service Profile Identifiers (SPIDs)	4-17
Profiles Which Require SPIDs	4-17
Definition of SPIDs	4-17
SPID Initialisation	4-18
SPID Debugging	4-19
Automatic Switch Detection	4-21
Call Control	4-22
Remote CAPI	4-26
Call Logging	4-27
Using a Domain Name Server	4-27
Slotted Interface Numbering	4-28
Always On/Dynamic ISDN (AODI)	4-28
Components of AODI	4-28
Configuring AODI	4-29
Data Over Voice	4-32
Configuration Examples	4-33
A Basic ISDN Setup	4-33
Refining the ISDN Setup	4-41
Command Reference	4-42
ACTIVATE ISDN CALL	4-42
ACTIVATE Q931 ASPID	4-43
ACTIVATE Q931 MESSAGE	4-43
ADD ISDN CALL	4-44
ADD ISDN CLILIST	4-50
ADD ISDN DOMAINNAME	4-50

ADD LAPD TEI	4-51
ADD LAPD XSPID	4-51
ADD LAPD XTEI	4-52
DEACTIVATE ISDN CALL	4-52
DELETE ISDN CALL	4-53
DELETE ISDN CLILIST	4-53
DELETE ISDN DOMAINNAME	4-54
DELETE LAPD TEI	4-54
DELETE LAPD XSPID	4-55
DELETE LAPD XTEI	4-55
DISABLE BRI CTEST	4-56
DISABLE BRI DEBUG	4-56
DISABLE BRI TEST	4-57
DISABLE ISDN CALL	4-58
DISABLE ISDN LOG	4-58
DISABLE Q931 DEBUG	4-59
DISABLE RCAPI	4-60
ENABLE BRI CTEST	4-60
ENABLE BRI DEBUG	4-61
ENABLE BRI TEST	4-62
ENABLE ISDN CALL	4-65
ENABLE ISDN LOG	4-65
ENABLE Q931 ASPID	4-66
ENABLE Q931 DEBUG	4-66
ENABLE RCAPI	4-71
RESET BRI	4-72
RESET BRI COUNTERS	4-72
RESET Q931	4-73
SET BRI	4-73
SET ISDN CALL	4-75
SET ISDN DOMAINNAME	4-80
SET ISDN LOG	4-81
SET LAPD	4-82
SET Q931	4-84
SHOW BRI CONFIGURATION	4-86
SHOW BRI COUNTERS	4-88
SHOW BRI CTEST	4-94
SHOW BRI DEBUG	4-95
SHOW BRI STATE	4-96
SHOW BRI TEST	4-100
SHOW ISDN CALL	4-103
SHOW ISDN CLILIST	4-107
SHOW ISDN DOMAINNAME	4-108
SHOW ISDN LOG	4-109
SHOW LAPD	4-110
SHOW LAPD COUNT	4-112
SHOW LAPD STATE	4-114
SHOW Q931	4-114
SHOW Q931 SPID	4-117

Introduction

This section describes the ISDN (Integrated Services Digital Network) service provided by the router, and how to set up and use ISDN on the router.

ISDN is defined by the ITU-T in a range of Recommendations. The principles of ISDN are stated in the ITU-T Recommendation I.120 (1988). The underlying principle is the support of a wide range of voice (telephone calls) and non-voice (data exchange) applications in the same network. This is done through the provision of a range of services using a limited set of connection types and user-network interface arrangements. These limitations serve to make international ISDN interconnection feasible. The primary application of ISDN is the provision of both circuit and packet switching, but ISDN also supports non-switched connections. The fundamental building block of ISDN is a 64 kbit/s switched digital connection.

The two most common methods for providing ISDN access at a customer's premises are called Basic Rate Access and Primary Rate Access. Basic Rate Access consists of two 64 kbit/s B channels and one 16 kbit/s D channel, whereas Primary Rate Access consists of up to 30 64 kbit/s B channels and one 64 kbit/s D channel.

The B channels are user channels, and carry digital data, PCM-encoded voice, or a mixture of lower rate traffic. All traffic on a B channel goes to the same destination, but each B channel may go to a different destination.

Three kinds of connections may be set up over a B channel:

- Circuit-switched—the circuit is set up by common channel signalling over the D channel (see below).
- Packet-switched—data is exchanged via a X.25 packet switching node.
- Semipermanent—the connection is set up by prior arrangement with the service provider. For more information about configuring the router to use semipermanent ISDN connections, see *Chapter 11, Time Division Multiplexing (TDM)*.

The D channel serves two purposes:

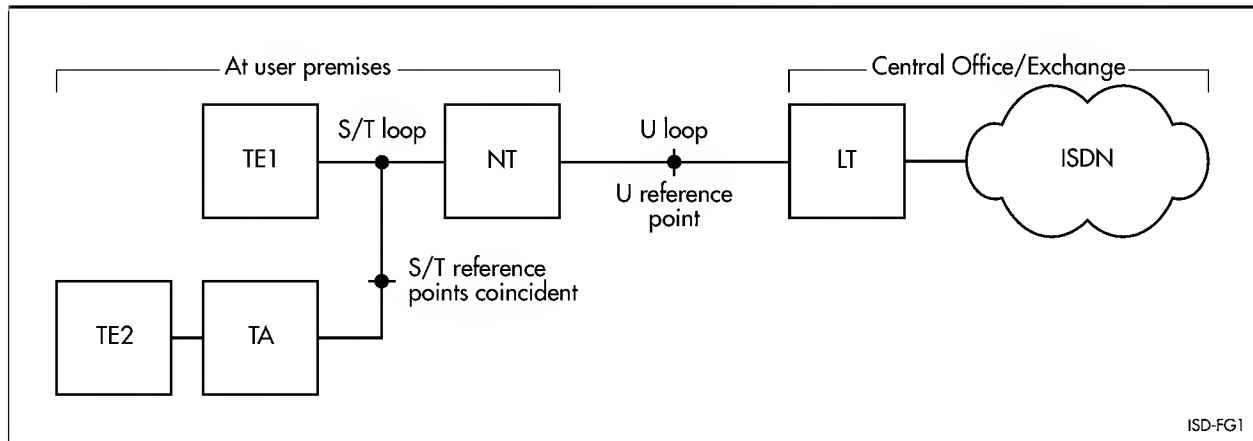
- Common channel signalling to control circuit switching.
- Low speed packet switching.

Basic Rate Access

A block diagram of a typical Basic Rate Access circuit is shown in Figure 4-1 on page 4-4. The router is classed as TE1 (*Terminal Equipment, type 1*). A TE2 is not directly compatible with ISDN and requires a *Terminal Adapter (TA)* so that it may make use of an ISDN. AR Series routers are all compatible with ISDN and do not require a TA for connection to the ISDN. The S/T loop portion of the circuit operates over a strictly limited distance and is intended for operation within customer premises. The S/T loop may be shared by a number of TE1s and TAs communicating with a single *Network Termination (NT)*. The U loop may be several kilometres in length and runs between the NT and the *Line Termination (LT)* on the ISDN service provider's premises. The letters S, T and U refer to reference points in the ITU-T Recommendations defining ISDN. In most countries the NT is provisioned by the ISDN service provider as part of the Basic Rate Access circuit. However, in the USA provision of the NT is the customer's responsibility. This has given the impetus to CPE suppliers to

integrate the NT into their equipment to avoid the requirement for a separate NT. The AR router family provides Basic Rate Interfaces for connection to either the S/T or U loops. The S/T interfaces may be used anywhere in the world (the customer may need to provide the NT in the USA), but the U interface may only be used in the USA. The characteristics of the two interface types are described below.

Figure 4-1: A typical ISDN Basic Rate Access circuit.



S/T Interfaces

Operation of the 4-wire S/T loop is defined in ITU-T Recommendation I.430. The S/T loop may be shared by more than one TE or TA, although there will usually be only one NT. There are a number of possible configurations for the TEs and the NT. The simplest is a point-to-point configuration where one NT communicates with one TE and a 100Ω termination resistor is connected across the receive and transmit pairs at each of the NT and TE. The short passive bus configuration is intended for use where up to 8 TEs are required to communicate with the NT. The TEs may be distributed anywhere along the passive bus which may be up to 200 metres in length. Termination resistors are located at the NT and at the other end of the passive bus, the TEs do not require termination resistors. An extended passive bus configuration comprises a group of TEs situated within 25 to 50 metres of one another on a bus that may be up to 500 metres long. As with short passive bus the termination resistors are located at the NT and at the other end of the bus, but not in the TEs. Branched passive bus is similar to extended passive bus, but in this case the termination resistors are located at the NT and just before the group of TEs at the opposite end of the bus, rather than at the very end.

Connection from the S/T loop to a TE is made via an RJ45 8-pin connector. The four center pins on the connector are used for the transmit and receive pairs. Power may be transferred from the NT to TEs (or vice-versa) over the signal wires or one of the outer pairs.

The 2B+D channels of the Basic Rate Access circuit require 144 kbits/s. However, once framing, synchronisation and other overhead bits are added, the total bit rate is 192 kbits/s. Data is transferred between the TEs and the NT in 48-bit frames, one frame every 250 microseconds. Each of these frames carries 4 D channel data bits and 16 bits for each of the B channels. Note the distinction between these frames used for communication between the TE and NT, and the HDLC frames used for user data transport over the B channels and for communication with the ISDN over the D channel. The HDLC frames are carried over the S/T loop frames.

Provision has been made in I.430 for additional communication channels for use between the TE and NT. Since these channels are synchronised by setting the M bit in every twentieth S/T frame their operation is called multiframing. There are 5 S channels in the NT to TE direction and one Q channel in the TE to NT direction. Each of these channels provides a data rate of 800 bit/s.

Since it is permissible to have more than one TE on an S/T loop there is a possible contention problem. The ISDN protocol ensures that a B channel is allocated to only one TE at a time, so contention for the B channels is resolved by the network. On the D channel, the LAPD addressing scheme (see “LAPD” on page 4-11) ensures that in the NT to TE direction data will reach its correct destination. However, in the TE to NT direction a mechanism is necessary to avoid transmission by two TEs at one time and to recover from situations where simultaneous transmission does occur. The details of this mechanism are beyond the scope of this discussion, but the essential elements are:

- The detection of collisions by TEs that are transmitting.
- One of the TEs involved in the collision will be able to complete its transmission successfully.
- A priority scheme to reduce collisions whereby the priority of a TE is reduced once it has completed a transmission until all other TEs have had a chance to transmit.

An additional feature of the priority scheme is the provision of two priority classes. The higher priority class is used for signalling information.

ITU-T Recommendation I.430 defines five transmission states for the S/T loop (Table 4-1 on page 4-5).

Table 4-1: S/T loop transmission states defined by ITU-T Recommendation I.430.

State	Meaning
INFO 0	No signal being transmitted.
INFO 1	TE transmits a continuous signal to wake up the NT.
INFO 2	NT transmits a continuous signal to wake up the TE, or in response to INFO 1 from the TE.
INFO 3	TE transmitting data, the fully operational state.
INFO 4	NT transmitting data, the fully operational state.

The circumstances under which each device transmits a particular INFO signal and the events which cause transmission to change, are determined by a state machine defined in I.430.

The usual transmission state for a TE and a NT at power on is INFO 0. Either of these devices may instigate a change to a higher state. This is known as activation. A higher layer in a TE can issue an activation request to the physical layer which, if it is in the deactivated state, will begin transmitting INFO 1 to try to wake up the NT. I.430 requires that the activation request time out through the use of a timer called T3 which has a maximum value of 30 seconds.

When a Basic Rate Access link is used to provide a semipermanent connection the activation and deactivation procedures may be disabled by the service provider. In this case the INFO 1 state is never entered, and the NT transmits INFO 2 by default and INFO 4 when it receives INFO 3 from the TE.

U Interfaces

In the USA, customer provided equipment is connected to the U loop; in all other countries the ISDN service provider will supply the NT. Operation of the NT is defined in the American National Standards Institute (ANSI) standard T1.601-1992. The 2-wire U loop may be not be shared by multiple NTs; it is a simple point to point link. Power is available on the U loop and the T1.601 standard specifies requirements for sealing current and DC metallic termination. DC and low frequency AC signalling formats are specified for initiating Insertion Loss Measurement and Quiet maintenance modes.

Data is transferred between the NT and the LT in 240-bit frames at a rate of one every 1.5 milliseconds. Each frame carries 96 bits for each B channel and 24 bits for the D channel. The remaining bits are used for synchronisation, an *Embedded Operations Channel* (EOC), CRC checking of the frames and the transfer of status bits between the NT and LT. The most important of the status bits are the “act” and “dea” bits which are used to control the activation and deactivation of the interface. Another bit, the “febe” bit, when set indicates that a CRC error in a frame transmitted by the NT has been detected by the LT. The quality of the transmission over the U loop can be monitored by counting the CRC errors detected by the NT and the CRC errors reported by the LT through the “febe” bit. Note the distinction between these frames used for communication between the NT and LT, and the HDLC frames used for user data transport over the B channels and for communication with the ISDN over the D channel. The HDLC frames are carried over the U loop frames.

When the NT is powered on the U interface will be in a deactivated state. The loop may be activated by either the NT or the LT. There is a defined procedure through which the loop is activated during which each end sets its echo cancellation parameters. This procedure may take as long as 15 seconds. Once both the NT and LT have synchronised to each other's signal the LT will change the “act” bit in its transmitted frames from 0 to 1. When the NT sees this change the activation process is complete. Unlike the S/T loop, which may be deactivated when there are no calls in progress, the LT will normally endeavour to keep the U loop active at all times. The LT is able to initiate a deactivation of the link by changing the “dea” bit in the frames it transmits to the NT from 1 to 0.

Support for ISDN

The ISDN Basic Rate S/T Interface (BRI) on the router conforms to ITU-T Recommendation I.430. The majority of the features required by I.430 are implemented by a specialised integrated circuit called the S/T transceiver. The BRI supports point-to-point, short and extended passive bus, and branched passive bus connection modes. The BRI is not powered from the NT, nor can it detect power from the NT. The router operates as a TE and does not offer TA functionality. The BRI is able to detect multiframing and indicate this to the manager, but the BRI does not make use of the Q or S data channels.

The BRI U interface on the router is only for use in the USA, and conforms to ANSI standard T1.601-1992. The U interface transceiver integrated circuit used is not the same for all U interfaces. Connection to the U loop is via an RJ45 8-pin connector using only the middle pair. The router's U interface does not take power from the U loop. The U interface meets the T1.601 sealing current and DC metallic termination requirements, as well as supporting the DC and low frequency AC signalling formats for initiating Insertion Loss Measurement and Quiet maintenance modes.

All BRI interfaces on the router support the automatic TEI assignment mode of operation.

BRI Physical Layer

The physical layer of the Basic Rate Interface (BRI) for the router is implemented in the BRI software module. The module requires no user configuration for normal ISDN operation. When used to support a semipermanent connection, some configuration is required. See below and *Chapter 11, Time Division Multiplexing (TDM)* for more information. Commands are provided to show the status of the module, and to examine and reset a number of data and error counters. The BRI module may be also be reset, but this should not be necessary during normal operation. A set of commands is also provided for testing the interface, but these should not be used during normal operation as they will interfere with the functioning of the router. Each command may specify the BRI interface on which it is to operate. For example:

```
SHOW BRI=0 STATE
```

shows the state of the first Basic Rate Interface. The BRI interface number is optional in some commands and if omitted, the command operates on all installed BRI interfaces.

When a layer 2 module (for example the Point-to-Point Protocol, PPP) wishes to use a BRI it *attaches* to the BRI module and specifies which slots it will be using. The BRI module then allocates a *channel number* to the layer 2 module for use when data is passed between the modules. In the following description, the B channels of the BRI are called slots and the groupings of slots being used by layer 2 modules are called channels. Data transferred over the BRI for each channel is encapsulated in HDLC frames.

Configuring and Controlling the Basic Rate Interface

The BRI software module does not require user configuration for normal ISDN operation, but the following command may be required when the interface is used for semipermanent connections:

```
SET BRI=n ACTIVATION={NORMAL|ALWAYS} MODE={ISDN|TDM|MIXED}  
[ISDNSLOTS=slot-list] [TDMLOTS=slot-list]
```

where *n* is the number of the BRI interface and must be specified. The ACTIVATION parameter controls the operation of the layer 1 state machine. The default is NORMAL and is the normal mode of ISDN operation. Setting ACTIVATION to ALWAYS indicates that the interface is connected to a link that is expected to be active at all times. When the link is not active the router will not attempt to activate the link by sending INFO 1. The MODE parameter determines whether the interface provides normal ISDN call functionality, or semipermanent connections, or a mixture of both. The default MODE for a BRI interface is ISDN and by default all of the slots are available for ISDN calls. The ISDNSLOTS parameter can be used to restrict the slots available for calls by specifying a list of eligible slots, effectively disabling some of the slots on a BRI link. If MODE is set to TDM the D channel is disabled and no ISDN calls can be made over the interface. See *Chapter 11, Time Division Multiplexing (TDM)* for more information about using an interface in TDM mode. When MODE is set to MIXED one slot may be used for an ISDN call and the other slot for a semipermanent connection.



*The **MODE** parameter of the **SET BRI** command affects the way the router behaves when connected to a network to the extent that, if configured inappropriately for the network to which it is connected, it may not conform to the national standards applying to that network. Therefore care must be taken when using this command. Please seek the advice of your distributor or ISDN service provider when changing the mode of operation from the default, which is the correct mode for connecting to a standard ISDN network.*



*Semipermanent connections are not available in the USA and the router will not permit the **MODE** of a **BRI U** interface to be set **TDM** or **MIXED** or the **ACTIVATION** mode set to **ALWAYS**.*

For example, to allow slot B1 to be used for an ISDN call, slot B2 to be used for a semipermanent connection and to disable the normal activation procedures, enter the command:

```
SET BRI=0 ACTIVATION=ALWAYS MODE=MIXED ISDNSLOTS=1 TDMSLOTS=2
```

In a slot list the numbers 1 and 2 correspond to slots B1 and B2, respectively.

The BRI software module and hardware may be reset with the command:

```
RESET BRI=n
```

where *n* is the number of the BRI interface. This command is not required for normal operation and should only be used under advice from your distributor or reseller.

To aid diagnosing TE/NT problems, debug messages generated as a result of certain events can be redirected to a port or to a Telnet session (Table 4-2 on page 4-8).

Table 4-2: Categories of debug messages generated by the BRI software module.

Category	Meaning
Errors	A BRI software module internal error.
Indications	An indication from the layer 1 state machine to a higher layer or the management layer.
State changes	A change of state for the layer 1 state machine.
Events	An event that is an input to the layer 1 state machine.

The commands:

```
ENABLE BRI [=instance] DEBUG [= {ERRORS | INDICATIONS | STATES |
EVENTS | ALL} ]
DISABLE BRI [=instance] DEBUG [= {ERRORS | INDICATIONS | STATES |
EVENTS | ALL} ]
```

allow a single debug option to be enabled or disabled on each invocation. However, successive commands can be used to disable or enable any desired combination of debug options. For example, the command sequence:

```
DISABLE BRI DEBUG=ALL
ENABLE BRI DEBUG=ERRORS
ENABLE BRI DEBUG=INDICATIONS
ENABLE BRI DEBUG=EVENTS
```

will enable the **ERRORS**, **INDICATIONS** and **EVENT** debug options on all BRI interfaces.

The command:

```
SHOW BRI DEBUG
```

displays the state of the debug categories.

The BRI module has several test modes that are used for testing the BRI hardware and for Telecommunication authority testing for standards conformance purposes. The commands:

```
DISABLE BRI=instance TEST[=test]  
ENABLE BRI=instance TEST=test
```

allow a single hardware test to be disabled or enabled on each invocation (see “*Command Reference*” on page 4-42 for a complete list of hardware test modes). However, any number of hardware tests may be run simultaneously by using successive commands to disable or enable particular hardware tests. For example, the command sequence:

```
DISABLE BRI=0 TEST  
ENABLE BRI=0 TEST=8  
ENABLE BRI=0 TEST=9
```

will enable hardware tests 8 and 9 on interface BRI0. The commands:

```
DISABLE BRI=instance CTEST  
ENABLE BRI=instance CTEST=ctest
```

allow the currently running conformance test to be disabled or a single specified conformance test to be enabled (see “*Command Reference*” on page 4-42 for a complete list of hardware test modes). Only one conformance test may be running at any one time.

The current conformance test modes may be viewed with the commands:

```
SHOW BRI TEST  
SHOW BRI CTEST
```



The TEST and CTEST modes are required for manufacturer testing only and should not be activated while the system is in normal use, as they will interfere with the functioning of the router.

Examining the Status of the Basic Rate Interface

The status of the BRI can be displayed with the command:

```
SHOW BRI STATE
```

For a BRI S/T interface the display shows:

- The operational mode of the interface: TE or NT.
- The state of the physical layer state machine: “Inactive”, “Sensing”, “Deactivated”, “Awaiting Signal”, “Identifying Input”, “Synchronized”, “Activated” or “Lost framing”.
- The received and transmitted INFO signals. In normal operation the BRI transceiver receives INFO 4 from the NT and transmits INFO 3.
- Whether or not an activation request is being processed, or the loop is activated.
- Whether or not the TE is synchronised to the NT.
- The activation mode of the interface: “normal” or “always”

- The mode of the interface: "ISDN", "TDM" or "mixed".
- The slots available for ISDN calls (only displayed when the interface is not in TDM mode).
- The slots available for TDM groups (only displayed when the interface is not in ISDN mode).
- The current D channel priority class, which may vary from one D channel frame to the next.
- Whether or not the B channels are attached to a higher layer module, and whether or not the B channels are aggregated.
- Whether the transceiver has detected multiframing in the data stream from the NT.
- The mask revision of the transceiver chip (on some hardware models).

For a BRI U interface the display shows:

- The operational mode of the interface: TE (or LT: test mode on some hardware models only).
- The state of the physical layer state machine: "Deactivated", "Activating", "Pending active", "Active" or "Pending deactivated".
- Whether or not an activation request is being processed, or the loop is activated.
- Whether or not the router is synchronised to the LT.
- The activation mode of the interface: always "normal".
- The mode of the interface: always "ISDN".
- The most recent EOC message received.
- The current maintenance mode: "none", "Quiet", "Insertion Loss Test Mode".
- The slots available for ISDN calls.
- Whether or not the B channels are attached to a higher layer module, and whether or not the B channels are aggregated.
- The mask revision of the transceiver chip (on some hardware models).

The command:

```
SHOW BRI CONFIGURATION
```

shows the higher layer modules (if any) that have been attached to the BRI interface. The display shows:

- The modules attached to the D, B1 and B2 channels.
- The bandwidth of the channel (for B channels only).
- A list of up to four addresses used to filter incoming frames on the D channel. The addresses are compared with the 16-bit field of the layer 2 frame which contains the SAPI and TEI for a D channel frame. The filter reduces the loading on the BRI software module by not interrupting it for frames which are intended for other TEs.
- An address mask which specifies which bits of an address are significant for comparison when filtering incoming D channel frames.

Monitoring Operation of the Basic Rate Interface

The BRI module provides a set of counters for monitoring the BRI interface. The counters are divided into 3 categories: interface counters, BRI counters and diagnostic counters. Counters from any of these categories can be displayed using the command:

```
SHOW BRI COUNTERS [= { INTERFACE | BRI } ]
```

If a category is not specified, all categories are displayed. If INTERFACE is specified, the counters from the interfaces table of the interfaces MIB relating to the BRI are displayed. If BRI is specified, counters relevant to a Basic Rate interface in particular, that are stored in the enterprise MIB, are displayed. The output has multiple sections, one for the BRI as a whole and one for each active channel. The meaning of each of the counters is described in “*Command Reference*” on page 4-42.

The counters in each category may be cleared to zero using the command:

```
RESET BRI COUNTERS [= { INTERFACE | BRI } ]
```

If a category is not specified, all counters are cleared.



Using the RESET BRI COUNTERS command on page 4-72 to clear the counters does not clear the MIB counters themselves. Instead, the contents of the MIB counters are copied to offset storage locations that are subtracted from the MIB counters before being displayed by the SHOW BRI COUNTERS command on page 4-88.

LAPD

LAPD is the Link Access Protocol for the ISDN D channel, as defined by ITU-T Recommendation Q.921. It is a layer 2, or data link layer, protocol which is used for communication between ISDN Terminal Equipment (TE, i.e. the router) and Network Equipment (NT, i.e. the ISDN exchange). LAPD is responsible for providing addressing, flow control, and error detection for higher layer users of the ISDN D channel. LAPD is similar to LAPB (layer 2 of X.25), with the addition of multiple logical connections, allowing a single D channel to support multiple layer 3 entities. LAPD is not used on ISDN B channels.

In normal operation the LAPD module will not require any configuring since the default configuration will allow it to function fully. The default for BRI interfaces is to operate with automatic TEI assignment.

Operation

The main purpose of LAPD is to provide Q.931 Call Control with a data link layer. Because Q.931 Call Control is mainly used when a call is being made or brought down there is a lot of spare bandwidth on the D channel. To allow this to be used LAPD can also operate as the data link layer for Q.931 Packet Mode and X.25 Packet Mode Operation. These modes allow the D channel to be used for the transfer of data, as well as for call control.

The LAPD parameters are specified by the LAPD standard and should not be changed.

Packet mode support

As mentioned above, LAPD can operate as the link layer for X.25 packet mode operation. Different ISDN profiles have different flavours of packet mode operation, but some of the ways in which packet mode operations are supported are given here.

Some ISDN switches in the USA and Canada require a fixed TEI for packet mode operations, even if data and voice calls are made using dynamic TEI allocation. To specify a fixed TEI for packet mode operation, use the command:

```
ADD LAPD=interface XTEI=tei
```

The TEI specified must be in the range 0-63.

If a fixed TEI is not required, and the router is required to perform SPID initialisation, packet mode operations must take place on the same TEI as a DLC which has a SPID which subscribes to the packet mode service. To specify which SPID subscribes to the packet mode service, use the command:

```
ADD LAPD=interface XSPID=spid-index
```

The SPID index specified is either "1" or "2", corresponding to the SPID1 and SPID2 parameters used in the SET Q931 command on page 4-84.

Fault Finding

The output from the SHOW LAPD command on page 4-110 can be useful when trying to find the cause of a fault in an ISDN link.

One possible problem involves obtaining a TEI from the network. A TEI is required for the D channel of each basic rate interface before a link can be established (the TEI for primary rate interfaces is always 0).

If the interface is set for automatic TEI assignment (which is the normal BRI setup) and an attempt has been made by the router to make a call then the SHOW LAPD command on page 4-110 should display a TEI for the interface (with a range of 64 to 126). If no TEI is present it means that the automatic TEI procedure is not operating.

The DLC parameter in the display can be used to check the state of each Data Link Connection (DLC, or logical link operating on the D channel). On both Basic Rate and Primary Rate interfaces there should be a SAPI of 63 for TEI management and a SAPI of 000 for Q.931 Call Control. The DLC for a CES of 001 is the DLC used to transport Q.931 Call Control information. If a call has been made on the ISDN interface then the state of this DLC should always be ALIVE. If it reads DEAD then the DLC for that interface can not be used for Q.931 signalling.

The SHOW LAPD STATE command on page 4-114 and the SHOW LAPD COUNT command on page 4-112 may be used to provide state and counter information about a LAPD interface.

Default Setup

The standard LAPD configurations are shown in Table 4-3 on page 4-13 (Basic Rate Interfaces) and Table 4-4 on page 4-13 (Primary Rate Interfaces).

Table 4-3: Standard LAPD configuration for an ISDN Basic Rate Interface.

Mode	Auto								
Debug	Off								
TEI	Provided by the network								
T, N and k values (for each SAPI):									
SAPI	Layer 3	T200	T201	T202	T203	N200	N201	N202	k
0	Q.931 Call Control	10	10	20	100	3	260	3	1
1	Q.931 Packet Mode	10	10	20	100	3	260	3	3
16	X.25 Packet Mode	10	10	20	100	3	1024	3	3
63	LAPD Management	10	10	20	100	3	260	3	1

Table 4-4: Standard LAPD configuration for an ISDN Primary Rate Interface.

Mode	nonAuto								
Debug	Off								
TEI	0								
T, N and k values (for each SAPI):									
SAPI	Layer 3	T200	T201	T202	T203	N200	N201	N202	k
0	Q.931 Call Control	10	N/A	N/A	100	3	260	N/A	7
1	Q.931 Packet Mode	10	N/A	N/A	100	3	260	N/A	7
16	X.25 Packet Mode	10	N/A	N/A	100	3	1024	N/A	7
63	LAPD Management	10	N/A	N/A	100	3	260	N/A	7

Addressing

The LAPD frame uses the HDLC frame format. The addressing function of LAPD allows multiple layer 3 entities to operate on one D channel and allows terminals on a BRI bus to be addressed. The 16-bit address in the HDLC frame is called the Data Link Control Identifier (DLCI). The DLCI is made up of a Service Access Point Identifier (SAPI), a Terminal Endpoint Identifier (TEI), and some additional control bits.

The SAPI determines the type of the layer 3 entity which is being addressed (Table 4-5 on page 4-14).

Table 4-5: SAPI values used by LAPD to specify types of layer 3 entities.

Value	Frame
0	Q.931 Call Control Information.
1	Q.931 Packet Mode Information.
16	X.25 Packet Mode Information.
63	LAPD Management Information.

The TEI indicates the specific logical device (in point-to-point connections) or a group of logical devices (in broadcast connections) within the individual SAP identified by the SAPI. TEI values are shown in Table 4-6 on page 4-14 (Basic Rate Interfaces) and Table 4-7 on page 4-14 (Primary Rate Interfaces).

Table 4-6: TEI values used by LAPD to specify logical devices attached to a Basic Rate Interface.

Value	Use
0	Reserved for NT2 equipment
1-63	Non-automatic assignment for TE equipment. The user assigns these.
64-126	Automatic assignment for TE equipment. The network assigns these.
127	All ones broadcast address.

Table 4-7: TEI values used by LAPD to specify logical devices attached to a Primary Rate Interface.

Value	Use
0	Used for all terminals.
1-126	Not used.
127	All ones broadcast address.

The Data Link Connection (DLC) is the name given for each valid combination of a SAPI and a TEI; each DLC is an individual logical link.

The Connection Endpoint Suffix (CES) is used by a layer 3 entity to identify individual DLCs within the layer 3 SAP.

Frame Control Fields

There are three types of LAPD frames (Table 4-8 on page 4-15).

I frames are used to transfer layer 3 data. Their control fields contain modulo 128 number sent and received counters to allow a window of unacknowledged frames to be sent before an acknowledge is received.

S frames are used by LAPD for link flow control. Their control fields only contain a number received count.

U frames provide additional data transfer or link control functions. They are used by LAPD for the transfer of management information.

Table 4-8: LAPD frame types.

Type	Use	Control Field Size
I	Numbered information frames	16 bits
S	Supervisory frames	16 bits
U	Unnumbered information frames	8 bits

Non-Associated Signalling

Normally, a given ISDN interface will use its own D channel for signalling for the calls that are made on the interface. However, it is possible to configure a mode of operation in which a given D channel provides the signalling for a number of ISDN interfaces. The advantage of this is that the D channels which are unused for signalling can then be used as B channels.

This feature is known as *non-associated signalling* or *common D channel*. The ISDN network must support the feature. At present the router will only support this feature if the Q.931 profile of participating interfaces is set to JAPAN. LAPD commands are used to set up the interfaces that are taking part in non-associated signalling, while Q.931 commands are used to give each interface a unique ID.

To set up an ISDN interface to be a master interface for non-associated signalling, use the command:

```
SET LAPD=instance NASMODE=MASTER
```

To set up an ISDN interface to be a slave interface for non-associated signalling, use the command:

```
SET LAPD=instance NASMODE=SLAVE NASMASTER=master-interface
```

where *master-interface* is the instance number or interface name of an ISDN interface whose NASMODE is MASTER.

To identify the ISDN interfaces for non-associated signalling, use the command:

```
SET Q931=instance INTID=hex-string
```

where *hex-string* is a sequence of hexadecimal digits which give the interface ID in hexadecimal. The interfaces operating in non-associated signalling mode and their interface IDs will be arranged by subscription to the ISDN provider. Note that the interface ID is a hexadecimal value; if the interface ID was, for example, the digit "0", the interface ID would have to be entered as INTID=30, since 30 is the hexadecimal value for the digit 0. The format of interface identifiers must be clearly understood and this information should be explicitly requested from the ISDN provider.

Q.931

Recommendation Q.931 and related recommendations from the ITU-T cover the network layer of Digital Subscriber Signalling System No. 1, which handles the user-network interface for control of ISDN calls. The Q931 module in the router implements the Q.931 protocol, on behalf of call control modules CC (for data calls), PBX (for voice calls) and X25T (for packet data calls).

There are many features and options available in the Q.931 protocol, and different network providers have implemented different flavours of Q.931. The router must be tested against a particular implementation and gain approval before it can be used in a particular network. The Q931 module contains all the functionality required to connect to a number of ISDN networks, but the particular network to which the router is connected must be specified, using the command:

```
SET Q931=interface PROFILE={5ESS|AUS|CHINA|DMS-100|ETSI|
JAPAN|KOREA|NI1|NZ}
```

The PROFILE parameter specifies which Q.931 implementation will run on a particular ISDN interface. The profile is set automatically whenever the router territory is changed by the SET SYSTEM TERRITORY command on page 1-56 of *Chapter 1, Operation*. The default territory is 'Europe' which sets the profile to ETSI.



If you are not sure which profile to use, contact your distributor or ISDN service provider.



Failure to select the correct profile will invalidate the approval of this product with respect to the applicable national standards for the country in which the product is used.

Other Q.931 parameters may be set using the command:

```
SET Q931=interface [timer={OFF|time}] [NONUM={ACCEPT|REJECT}]
[NOSUB={ACCEPT|REJECT}] [NUM1=number] [NUM2=number]
[RATE={56K|64K}] [SPID1=spid] [SPID2=spid]
[SUB1=subaddress] [SUB2=subaddress]
```

As an aid to resolving Q.931-related problems, Q.931 debugging messages may be enabled or disabled with the commands:

```
ENABLE Q931=interface DEBUG={MDECODE|MRAW|SDLC|SINTERFACE|
SSPID|SSPIDFILE|STATE|TRACE}
DISABLE Q931=interface DEBUG={MDECODE|MRAW|SDLC|SINTERFACE|
SSPID|SSPIDFILE|STATE|TRACE}
```

The MRAW and MDECODE options display Q.931 messages sent or received via the specified ISDN interface, on the terminal from which the command was entered. The MRAW option displays a raw dump of the entire message, as a hexadecimal representation of the octets of the message. The MDECODE option displays a partially decoded version of the message. The call index, message type and information elements (IEs) in the message are all displayed, along with a raw dump of the contents of each IE.

The TRACE option provides a full trace of all subroutines executed in the Q931 module. This option is intended for use by router development and customer service engineers only.

The other options provide display of the various state machines in the Q931 module. The STATE option provides state and event debugging for ISDN calls. The SSPID option provides state and event debugging for the SPID state machine. The SSPIDFILE option provides state and event debugging for the SPID file state machine. The SDLC option provides state and event debugging for the DLC state machine. The SINTERFACE option provides state and event debugging for the interface state machine.

A Q.931 interface, an active call, or all active calls on an interface may be reset with the command:

```
RESET Q931=interface [CALL={call-index|ALL}]
```

where *call-index* is the index of an active Q.931 call. A RESTART message for the interface or call(s) is sent to the network. The specified call index must be the index for Q.931, not for call control. To display a list of Q.931 calls, use the command:

```
SHOW Q931 CALL
```

Service Profile Identifiers (SPIDs)

A feature of Basic Rate ISDN in the US and Canada is the requirement for the TE (that is, the router) to initialise before making calls. Initialisation consists of registering a Service Profile Identifier (SPID) with the switch to which the router is connected. The router sends the SPID to the switch in an INFORMATION message, and if the switch accepts the SPID, it sends an INFORMATION message back with the endpoint identifier that identifies the router in future call setup and disconnection.

The main points of SPIDs and SPID initialisation are as follows:

- Only certain profiles require the router to perform SPID initialisation, typically those for use in the USA and Canada.
- Valid SPIDs can be set for the router in a number of ways, including manual entry and automatic notification from the switch.
- SPID initialisation takes place every time the router is given a new TEI on a given DLC. A different SPID is required for each DLC.
- SPIDs are a sequence of decimal digits. Typically, the SPID includes the directory number.
- The router provides extensive debugging and monitoring facilities to help track SPID initialisation.

Profiles Which Require SPIDs

The profiles which require SPID initialisation are the Basic Rate profiles NI1, 5ESS and DMS-100. The profile AUS for Australian Basic Rate will use SPIDs if SPIDs are defined manually. This provides support for the Spectrum service in Australia, which runs on DMS-100 switches.

Profiles that do not allow SPIDs will go directly to the SPID OP state from the INIT state. Profiles that require SPID initialisation will make transitions in the SPID state machine based on the SPIDs defined.

Definition of SPIDs

SPIDs can be defined in a number of ways, only some of which are related to management commands. For this reason, SPIDs are not stored as part of the router configuration, but in separate SPID files. The SPID files contain all the SPID information for a single DLC on a Basic Rate interface.

The command:

```
SET Q931=interface [SPID1=spid] [SPID2=spid]
```

sets manual SPIDs. The command:

```
SET Q931=interface [NUM1=number] [NUM2=number]
```

sets generic SPIDs, in the case where the number consists of 10 digits. A generic SPID consists of a 10 digit directory number (3 digit area code and 7 digit local number) suffixed by the digits "0101".

SPIDs can also be defined via the auto-SPID mechanism. The router sends an INFORMATION message to the switch containing the universal SPID (the string "010101010101"). If the switch supports the auto-SPID procedure, it will respond with a sequence of INFORMATION messages containing valid SPID values. The router can select one of these SPIDs in certain circumstances, or store the SPIDs for display via the command:

```
SHOW Q931=interface SPID
```

A SPID can be selected for use with the command:

```
ENABLE Q931=interface ASPID=index
```

The router will automatically select a SPID when the switch presents only one or two valid SPIDs. Since the router can operate with either one or two SPIDs, in both cases the router will save the SPIDs presented and proceed to attempt SPID initialisation.

At any time the whole auto-SPID procedure can be restarted with the command:

```
ACTIVATE Q931=interface ASPID
```

This command will delete all existing auto-SPID information and initiate another request for auto-SPID information. If the router had already initialised with a manual or generic SPID and the auto-SPID request fails, the router will revert to the manual or generic SPID.

SPID Initialisation

Every time a given DLC is assigned a TEI, SPID initialisation must take place on that DLC. In normal operation, a TEI will be assigned for a DLC when the router first starts up, and this TEI will remain while the router is active. SPID initialisation takes place by the router sending an INFORMATION message containing the SPID currently defined for the DLC. This SPID is taken from the SPID file, and depending on the previous sequence of SPID initialisation and commands entered may be a manual SPID, a generic SPID, a SPID selected via the auto-SPID procedures, the universal SPID, or no SPID at all. The SPID file state machine keeps track of all previous SPID operations. The SPID file state can be seen in the output of the command

```
SHOW Q931=interface SPID
```

During operation, it is possible for a given TEI to be removed and a new one assigned. This is not a normal situation, and is usually due to communication being lost between the router and the switch at a lower layer. When a new TEI is assigned, SPID initialisation must take place again before calls can be made from the router.

SPID Debugging

The process of SPID assignment and initialisation is one of the most problematical in connecting devices to Basic Rate ISDN. Because of this, a number of debugging facilities have been provided to help the process. To enable debugging of the SPID initialisation process use the command:

```
ENABLE Q931=interface DEBUG=SSPID
```

This command will display the events and state transitions of SPID initialisation to the device from which the command was entered. The SPID states are given in Table 4-9 on page 4-19. The SPID events are given in Table 4-10 on page 4-20.

Table 4-9: SPID Initialisation States.

State	Description
NULL	Initial state for the SPID state machine at router restart.
IWAIT1	Router has sent specific SPID and is waiting for response from the network.
IWAIT2	The network has sent a prompt for SPID initialisation and the router has replied.
IWAIT3	The router has previously performed SPID initialisation, has seen a network prompt for SPID initialisation and has replied.
AWAIT1	The router is attempting auto switch detection and has sent a specific SPID.
AWAIT2	The router is attempting auto switch detection, has sent a Protocol Version Control message to the network and is waiting for a response.
AWAIT3	The router is attempting auto switch detection, has seen a prompt for SPID initialisation from the network and has replied.
5ESSNOTINIT	The router profile is 5ESS and an initialisation request has been sent to the network.
ASPID1	The router has sent the universal SPID (for auto SPID procedures) and is waiting for a response from the network.
ASPID2	The router has seen a network congestion message and is waiting (for 10 minutes) to restart auto SPID procedures.
ASPID3	The router has seen a number of auto SPID values and is waiting for user intervention to select the correct SPID(s).
ASPID4	The router is attempting auto switch detection, has sent the universal SPID (for auto SPID procedures) and is waiting for a response from the network.
OP	SPID initialisation has successfully taken place and normal operation can begin.
5ESSPINIT	The router profile is 5ESS and the router has initialised for point-to-point operation.
5ESSMINIT	The router profile is 5ESS and the router has initialised for point-to-multipoint operation.

Table 4-10: SPID Initialisation Events.

Event	Description
ASD	Perform auto switch detection.
INIT	Initialise.
TSPID	SPID timeout.
INFO	Received an INFORMATION message containing SPID information.
DLRELEASE	The LAPD data link has been released.
RESET	Reset the SPID state machine.
MIM	SESS management information message.
RELCOMP	Received a RELEASE COMPLETE message.
MESSAGE	Received a call control message which has implications for SPID initialisation.

SPID information is stored in SPID files. SPID file states are defined to control which of manual, generic and auto-SPID information are actually used in SPID initialisation. To enable debugging of the SPID file state machine, use the command:

```
ENABLE Q931=interface DEBUG=SSPIDFILE
```

The states and events for the SPID file state machine are given in Table 4-11 on page 4-20 and Table 4-12 on page 4-21 respectively.

Table 4-11: SPID File States.

State	Description
1	No SPIDs entered, auto SPID not run or in progress.
2	Manual SPID last one entered.
3	Generic SPID last one entered.
4	Auto SPID successful.
5	Auto SPID failed (non-initialising terminal).
6	Manual or generic SPID failed (non-initialising terminal).
7	Manual SPID after successful auto SPID.
8	Generic SPID after successful auto SPID.
9	Switch only supports non-initialising terminal.
10	Manual SPID passed, auto SPID initiated.
11	Generic SPID passed, auto SPID requested.
12	Manual SPID passed.
13	Manual SPID passed, generic SPID entered.

Table 4-12: SPID File Events.

Event	Description
SetSPID	The user has configured a SPID with the SET Q931 command on page 4-84.
SetDN10	The user has configured a 10 digit directory number with the SET Q931 NUM1/2 command
AutoSPIDPass	The auto SPID has been used for initialisation and initialisation has succeeded.
AutoSPIDFail	The auto SPID has been used for initialisation and initialisation has failed.
ConfSPIDPass	The configured SPID has been used for initialisation and initialisation has succeeded.
ConfSPIDFail	The configured SPID has been used for initialisation and initialisation has failed.
ConfSPIDTimeout	The configured SPID has been used for initialisation and the TSPID timer went off.
SPIDInit	The SPID file has been reinitialised.
NITIndication	An indication has been received that the router has to operate as a non-initialising terminal.
SetAutoSPID	The user has requested auto SPID procedures be retried.
ClearSPID	A manually configured SPID has been cleared.

Automatic Switch Detection

The router can, for Basic Rate interfaces in the USA and Canada, automatically detect the type of switch to which it is connected. This process is automatically initiated at router start-up when the router's personality PROM indicates that the router is manufactured for the USA market. The results of automatic switch detection are stored in a file whose name has the format:

```
BRIn.ASD
```

where *n* is the interface index. The automatic switch detection process can be debugged with the command:

```
ENABLE Q931=interface DEBUG=SINTERFACE
```

The interface states and events are given in Table 4-13 on page 4-21 and Table 4-14 on page 4-22.

Table 4-13: Automatic Switch Detection States.

State	Description
ASD-0	Initial state for the auto switch detection state machine at router restart.
ASD-1	Auto switch detection has been initiated by resetting the physical layer.
ASD-2	A TEI has been assigned at the LAPD layer and a data link establish requested.
ASD-3	The data link has established and an ASD event has been sent to the SPID state machine.

Table 4-13: Automatic Switch Detection States.

State	Description
ASD-4	The first SPID ASD event timed out and we have reset the physical layer again.
ASD-5	A TEI has been assigned at the LAPD layer again and a data link establish requested.
ASD-6	The data link has established again and an ASD event has been sent to the SPID state machine.
Operational	The interface type has been established and SPID initialisation can proceed.

Table 4-14: Automatic Switch Detection Events.

Event	Description
ASD request	Request to being auto switch detection.
Set profile	The interface type (profile) has been manually set.
DL-Establish	The data link layer has established.
5ESS msg	A message identifying the network as a 5ESS custom switch has been received.
SPID timeout	The SPID procedures have timed out.
ASD valid	The SPID state machine has been able to determine what sort of switch the router is attached to.
TEI assign	LAPD has assigned a TEI for the interface.
TEI remove	LAPD has removed the TEI for the interface.
DL-Release	The data link layer has been released.

Call Control

ISDN call control is responsible for maintaining and controlling ISDN calls. The call control module uses Q.931 to set up and tear down ISDN calls. Call control provides the interface between modules (such as PPP) that wish to use ISDN to send data, and the modules that directly control ISDN in the router.

In the description of ISDN call control, a distinction is made between an active call and a call definition. A call definition contains the configurable details of an ISDN call. Call definitions are modified by commands to configure the way that the router makes and responds to actual ISDN calls. An active call is an actual ISDN call. Each active call is the result either of a call definition being activated, or of an incoming call that has been matched to a call definition.

Before the router can make or accept ISDN calls, at least one call definition must be configured. Depending on the type of call configured, user modules, such as PPP, may also need to be attached to the call definition. The configuration of the call definitions will determine the behaviour of all ISDN calls in the router. To allow flexibility in a large number of situations, the call definition has a large number of possible options, and many ways of achieving the same result, of connecting two routers with an ISDN call.

The call definition serves two basic functions; to define how a router makes ISDN calls, and to define how a router receives ISDN calls. For an ISDN call to be made successfully between two routers, the active call on each router must be associated with a call definition. For this reason, call definitions may end up being defined in pairs on the two routers that are to communicate, with each call definition referencing information associated with the other call definition.

Two basic models of operation of call definitions are available on the router. In the first, each call definition is linked in some way with a call definition on another router. The call definition option description below details how the calls may be linked. Each call definition in this model is usually configured to be attached to by a higher layer protocol, and the higher layer instances are created before the call is activated.

In the second model of operation, a single call definition is set up on a router which is able to receive a large number of calls from different routers. The call definition is configured to extract some portion of the incoming SETUP message and use it to provide identification of the remote router. The remote router identification is used to configure higher layer modules and to dynamically create interfaces. This model provides a good way to allow a large number of remote routers to call a single central router, without having to create a large number of call definitions on the central router.

As there are a large number of options for call definitions, it is important to understand those options that relate to the situation in which the router will be used, both with respect to the model of operation and to the actual ISDN network being used. Having determined the best way to set up ISDN call definitions for a particular situation, it is advisable to use similar call definitions for all calls.

The following paragraphs outline the options of call definitions in broad groups.

Outgoing SETUP parameters specify the format and content of the SETUP messages originated by the router when it is making a call. To allow successful connection between routers, information must be carried in the SETUP message that can be interpreted at the remote router. Information elements in the SETUP message can be used to carry this information. The router carries information in three different information elements, the user-user data IE, the called subaddress IE and the calling party number IE. Each of these can be independently configured to carry the required connection information. The user-user data IE and called subaddress IE can be configured to carry the local call name or the remote call name. The called subaddress IE can also be configured to carry an arbitrary string of digits. The calling party number IE can be configured to carry the calling number of the call, the number of the Q.931 interface that the call uses, or to carry no number, which will then be supplied by the network.

A router receiving an ISDN call must have some way of identifying and checking the call. *Searching and checking parameters* in the call definition control this function. A call definition can be configured to search on the incoming call's user-user data IE, called subaddress IE or calling party number IE. Calls can also be set up to respond to any incoming call. As with the outgoing SETUP parameters, user-user data and called subaddress IEs can be compared with the call name or the remote call name. The following procedure is used to associate an incoming ISDN call with a call definition:

1. If the incoming call SETUP message contains a called subaddress IE, search all call definitions that allow searching on the called subaddress IE for a call

definition with a call name or remote call name matching the contents of the called subaddress IE in the call SETUP message. If a match is found, use the matching call definition to handle further processing of the call. Otherwise, go to step 2.

2. If the incoming call SETUP message contains a user-user data IE, search all call definitions that allow searching on the user-user data IE for a call definition with a call name or remote call name matching the contents of the user-user data IE in the call SETUP message. If a match is found, use the matching call definition to handle further processing of the call. Otherwise, go to step 3.
3. If the incoming call SETUP message contains a calling party number IE, and the IE contains calling party number digits, search all call definitions that allow searching on the calling party number IE for a call definition with a called number matching the contents of the calling party number IE in the call SETUP message. If a match is found, use the matching call definition to handle further processing of the call. Otherwise, go to step 4.
4. Search for a call definition configured to match any incoming call SETUP message. If a match is found, use the matching call definition to handle further processing of the call. Otherwise, reject the call.

Once identified, an ISDN call can also be checked. Checks can be made against the user-user data and called subaddress IE, as well as the calling party number IE. Calling party number information is also known as CLI (calling line information). CLI provides the greatest number of options as well as the greatest security, because the CLI is verified by the ISDN and cannot be falsified. An ISDN call in the router can be set up to require that CLI be present and that the number in the CLI be in a configured list of numbers.

Call precedence is used to resolve call collisions. These occur when two routers attempt to make a call to each other at the same time, and the call definition at each end is associated with an outgoing and incoming call simultaneously. The precedence parameter in the call definition will be used to determine which active call is cleared and which is accepted. Call precedence must be set IN on one router and OUT on the other for this scheme to work.

Call tenacity refers to the ability of the router to retry ISDN calls that fail. Calls may be retried as a series of retry groups. Each retry group consists of a series of retries. The time between retries and retry groups and the number of calls in a retry group and the number of retry groups may all be specified. An alternate number to try may also be specified. This will be used when all retries and retry groups have been tried and failed. A separate parameter specifies that a call is to be held up at all costs, so that it will be retried even when all retries have failed.

The *required* or *preferred* ISDN interface for a call to use may be specified. If the required interface is specified, the call may only be made on that interface. If the preferred interface is specified, the call will be tried on that interface first. In either case, the call will only be tried on an interface that has a free channel.

The *call holdup* facility ensures that a call, once established, is held active for a specified minimum period of time. This ensures that the maximum benefit is obtained for calls made over a network that has a minimum call charge.

The alternate number facility gives the network manager the option of defining an alternate ISDN number for any ISDN call, which is independent of the main ISDN number. If a call to the main number fails, the alternate number, if defined, will be used to make a backup call, with the following restrictions:

- ISDN call retry parameters (RN1, RN2, RT1, RT2) apply only to the main number, not to the alternate number. The alternate number is tried only once.
- If call retry parameters are defined in such a way as to ensure that the main number is actually retried, the alternate number is not used until all retries have been tried and have failed.
- The KEEPU parameter, if set to TRUE, ensures that the call is retried from scratch. That is, the main number will be tried and retried and then the alternate number will be tried. The effect of the KEEPU parameter is checked only after the alternate number has been tried and failed.

This combination of flags and parameters ensures that a flexible combination of retries of the main and alternate numbers is achievable.

The *call back* facility enables a call to be configured to call back the originator of an incoming call.

Call bumping makes use of call priorities assigned to voice and data calls to terminate an existing active call in favour of a new call with a higher priority. For voice calls the priorities allowed are NORMAL and HIGH. For data calls, the priority is a number in the range 0-99. The rules for call bumping are:

- Call bumping takes place when all B channels on a given ISDN interface are in use and a new incoming or outgoing call is made.
- A high priority voice call is never bumped.
- A normal priority voice call can only be bumped by a high priority voice call.
- Data calls are bumped according to their priority (Table 4-15 on page 4-25).

Table 4-15: Call priority and call bumping.

Calls of priority...	Are bumped by...
0-19	Incoming or outgoing voice calls and incoming or outgoing data calls of higher priority.
20-39	Outgoing voice calls and incoming or outgoing data calls of higher priority.
40-59 (including the default priority of 50)	High priority outgoing voice calls and incoming or outgoing data calls of higher priority.
60-99	High priority outgoing voice calls and outgoing data calls of higher priority.



Although Table 4-15 specifies that a call can be bumped by an incoming call, it is likely that the ISDN to which the router is attached will not offer another incoming call if all B channels are in use. Instead a busy signal will be returned to the originating caller.

Calls are added (defined) and deleted with the commands:

```
ADD ISDN CALL=name NUMBER=number PRECEDENCE={IN|OUT}
options...
DELETE ISDN CALL=name
```

A call definition can be modified with the command:

```
SET ISDN CALL options...
```

Calls are enabled and disabled with the commands:

```
ENABLE ISDN CALL=name  
DISABLE ISDN CALL=name
```

Calls are made and disconnected with the commands:

```
ACTIVATE ISDN CALL=name  
DEACTIVATE ISDN CALL=name
```

The command:

```
SHOW ISDN CALL
```

displays information about call definitions and active calls.

Remote CAPI

The *Common Application Programmer's Interface* (CAPI) enables an ISDN client to control one or more ISDN devices. Typically, the client is an application on a PC and the ISDN device is an ISDN telephone, facsimile or modem connected to the PC. Remote CAPI is an extension to CAPI for controlling ISDN devices over a network, using a control protocol. The router supports RCAPi using RVS Datentechnik's proprietary *Device Control Protocol* (DCP).

RCAPi is enabled and disabled using the commands:

```
ENABLE RCAPi  
DISABLE RCAPi
```

RCAPi is disabled by default. When RCAPi is enabled, ISDN call control is managed by the RCAPi client. All data received on the B channel is passed to the PC and all data from the PC is transmitted on the B channel. Both HDLC framing and bit transparent data transfer modes are supported, enabling both voice and facsimile calls to be managed by the RCAPi client.

RCAPi coexists with the router's own call control for data and voice calls. For outgoing calls originating from the router, the voice and data functions on the router will use the router's own ISDN call control. Outgoing RCAPi calls have the same call bumping privileges as normal voice calls. The handling of incoming voice and data calls depends on the type of call and the state of RCAPi:

- If RCAPi is disabled or there are no active sessions with a listen request active, an incoming call will be passed to voice or data elements in the router, if the bearer capability and other information elements are compatible.
- Otherwise, the incoming call will be presented to all active RCAPi sessions with a listen request active.
- If one of the listening RCAPi sessions accepts the call, the call will proceed to that RCAPi session.
- If all listening RCAPi sessions reject the call, the call will be passed to the voice and data elements in the router, if the call's bearer capability and other information elements are compatible.

Call Logging

A call logging facility records details of events associated with ISDN calls. Log entries are sorted according to the time the call was initiated. Call logging is enabled or disabled with the commands:

```
ENABLE ISDN LOG
DISABLE ISDN LOG
```

An entry is added to the log when a call is initiated. When the log exceeds a predefined maximum length, the oldest entry that is in the CLEARED state is removed from the log. If no entries qualify the log is allowed to grow larger than the maximum defined length. Log messages can be sent to an asynchronous port on the router when the log entry enters the CLEARED state. The maximum length of the log and the port to which messages should be sent can be set with the command:

```
SET ISDN LOG [PORT={0..23|NONE}] [LENGTH=0..100]
```

Setting the PORT parameter to NONE disables the forwarding of messages to an asynchronous port. The default value for PORT is NONE. The default for LENGTH is 25. Call logging is enabled by default.

The command:

```
SHOW ISDN LOG
```

displays the current contents of the call log.

In addition to the call logging facility, the following events associated with ISDN calls are logged to the routers logging facility:

- Call activated.
- Call disconnected after normal call clearing.
- Call cleared due to an error condition.

For more information about the storage and display of these log messages, see *Chapter 12, Logging Facility*.

Using a Domain Name Server

For calls designed to carry IP traffic, an IP address is required. A *Domain Name Server* (DNS) can be used to determine the IP address for individual users. A domain name can be defined using the command:

```
ADD ISDN DOMAINNAME=domain-name
```

When a user logs in to the router, the user's login name is prepended to the domain name and a DNS lookup is performed using the resulting string. If the lookup is successful, the response is used as the IP address for the user.

The domain name may be deleted using the command:

```
DELETE ISDN DOMAINNAME
```

The currently assigned domain name can be displayed with the command:

```
SHOW ISDN DOMAINNAME
```

Appropriate entries must be created in the DNS to map entries of the form *login-name.domain-name* to IP addresses.

Slotted Interface Numbering

BRI interfaces are collectively termed “slotted” interfaces, in reference to their 64 kbit/s slot-based channel structure. For each slotted interface operating in ISDN mode there will be a LAPD and Q931 module instance. These instances are identified by the index of the slotted interface. For example, if there is a BRI interface on the base board of the router, then instance 0 of the LAPD and Q931 modules will correspond to the BRI0 interface.

Slotted interfaces that are operating in TDM mode do not need LAPD or Q931 module instances, so when a slotted interface is set to TDM mode the corresponding LAPD and Q931 instances are destroyed. Any remaining instances are not renumbered. Following the example above, if the BRI0 interface is set to TDM mode LAPD instance 0 and Q931 instance 0 are destroyed. LAPD instance 1 and Q931 instance 1 corresponding to the PRI0 interface are unaffected.

Always On/Dynamic ISDN (AODI)

Always On/Dynamic ISDN (AODI) is a networking service which provides a connection to TCP/IP-based services which is always available, and which adjusts bandwidth as required by the traffic traversing the connection. The defining document for AODI is “*Always On/Dynamic ISDN*” by A. Kuzma, written for the Vendor’s ISDN Association (<http://www.via-isdn.org/>).

AODI uses the ISDN D channel X.25 packet service to maintain a permanent connection between the end-user router and the Internet provider. This provides a constant, low-cost connection for low bandwidth requirements such as sending and receiving Email, news feeds, etc. When additional bandwidth is required, for example for web browsing, AODI automatically adds circuit switched connections over the ISDN B channels. When the additional bandwidth is no longer required, the B channels are dropped while the X.25 service remains.

Components of AODI

AODI is not a distinct protocol, but a service comprising features from other protocols, which when configured correctly, combine to provide the AODI service. The components of AODI are:

- D channel support for X.25 packet mode.
- X.25 support for PPP.
- ISDN B channel support for PPP.
- PPP support for multilink, bandwidth-on-demand and BAP.

D Channel Support for X.25 Packet Mode

The main purpose of the ISDN D channel is to act as a path for ISDN call control. The Q.931 protocol uses the D channel to send message to, and receive

messages from, the network in order to make and tear down calls. As an optional feature, the D channel can also be used to make X.25 packet mode connections to other X.25 devices on the ISDN, or other X.25 devices on packet mode networks. A device makes connections to the ISDN on the D channel by opening Data Link Connections (DLCs) on the D channel. A given DLC can be used by either Q.931 or X.25 for communicating with the network. However, in some cases, extra configuration may be required to allow X.25 to communicate.

In most countries, only one DLC will be opened on the D channel. This will be used by X.25 if X.25 is configured to use the D channel. In the USA, the DLC to be used by X.25 must be specified. A given TEI (which identifies the DLC) may be used (in which case a new DLC will be opened), or an existing DLC selected by specifying which SPID is available for X.25 use. Identifying the SPID will identify the DLC.

X.25 Support for PPP

The router supports MIOX (*Multiprotocol Interconnect Over X.25*), which allows different higher layer protocols to use an X.25 service. When an X.25 call is made, the data in the call setup message specifies which higher layer protocol is to be run over the call. One value for this data has been specified to indicate that the higher layer protocol is to be PPP.

By configuring MIOX and PPP correctly, a PPP link can be established over an X.25 interface. The AODI specification mandates that only X.25 switched virtual circuits (SVCs) are to be used for AODI.

ISDN B Channel Support for PPP

ISDN B channels are available for use by a number of different traffic types, including voice and data. The router allows PPP to run over ISDN B channels, controlled by ISDN call control call definitions. Calls can be configured for outgoing and incoming access, with a large number of options.

PPP Support for Multilink, Bandwidth-on-demand and BAP

AODI requires that the separate PPP connections on the D channel and on the B channels be joined together as a PPP multilink bundle. This allows the effective bandwidth of a PPP interface to be increased by making a number of different connections, and logically joining them together. The addition and removal of PPP links from the multilink bundle is controlled by BAP (*Bandwidth Allocation Protocol*).

The PPP links running on the B channel must be configured to run as bandwidth-on-demand links, otherwise when the PPP interface is configured, all links will be brought up at once. Bandwidth on demand means that certain links are only brought up when the bandwidth requirements of the multilink bundle demand it.

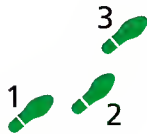
Configuring AODI

The following example illustrates the steps required to configure the router to communicate with an ISP using AODI. Only the configuration of the router is shown. The configuration shown is for a USA ISDN, to illustrate the extra commands for USA X.25 packet mode. Table 4-16 on page 4-30 lists the configuration parameters.

Table 4-16: Example configuration parameters for AODI.

Parameter	Value
Router IP address/mask	202.36.163.55/255.255.255.0
ISP X.25 DTE address	1234567890
Routers X.25 DTE address	1243452345
ISP ISDN number	1234567899
TEI for X.25 packet mode	21

In this example, the router has a single ISDN interface, BRI0. This is also referred to by its interface index, 0. The system territory and Q.931 profile are assumed to be set correctly for this router.



To configure AODI:

1. Configure X.25 packet mode TEI on LAPD.

A TEI has been supplied by the ISDN service provider for the use of X.25 on LAPD. To enter this information into the LAPD configuration, use the command:

```
ADD LAPD=0 XTEI=21
```

This ensures that when X.25 attaches to LAPD, LAPD will assign a DLC with a TEI of 21. This is required because the network will be expecting X.25 messages on TEI 21.

The setting of a TEI for use by X.25 is a network dependent option. The only currently known profiles which may require the TEI to be set are the USA profiles NI1, 5ESS and DMS-100. Contact your ISDN service provider for more information.

2. Configure X.25 to use LAPD.

An X.25 DTE interface must be created to use LAPD, using the command:

```
CREATE X25T=0 OVER=LAPD0 DTE=1243452345
```

The DTE address specified is the router's own DTE address. The ISDN service provider will provide this information.

3. Configure a MIOX circuit for use by PPP.

A MIOX circuit must be configured to allow X.25 calls to be made to the correct remote DTE address. This circuit can then be used by higher layers, including PPP:

```
ADD MIOX=0 CIRC=AODI DTE=1234567890
```

The DTE specified is the DTE address of the router at the remote end of the link. The ISDN service provider will provide this information.

4. Configure an ISDN call for use by PPP.

To allow PPP to make calls on the B channels of the ISDN interface, ISDN call definitions must be created. These call definitions specify the remote number to call, and how the call is to be identified to the remote router. In this case, the remote router has Caller Line Identification (CLI) enabled, and the network will present the caller's number (this router's number) without any configuration being required:

```
ADD ISDN CALL=AODI NUM=1234567899 PREC=OUT
```


5. Configure PPP to use the MIOX circuit and ISDN call.

Having created the underlying links for PPP, the PPP interface itself can be configured. The primary link will be over the MIOX call, while the B channels will be configured as demand links:

```
CREATE PPP=0 OVER=MIOX0--AODI
ADD PPP=0 OVER=ISDN-AODI TYPE=DEMAND NUMBER=2
```

6. Configure bandwidth parameters on the PPP interface.

The bandwidth parameters on a PPP interface allow the user to configure conditions under which extra bandwidth will be requested, and excess bandwidth removed. The parameters are UPRATE (the percentage utilisation above which extra bandwidth is requested), UPTIME (the time in seconds for which the excess utilisation must be present), DOWNRATE (the percentage utilisation below which excess bandwidth is removed) and DOWNTIME (the time in seconds for which the lower utilisation must be present). The default values for these parameters are 80%, 30s, 20% and 60s respectively.

These value do not work very well for AODI because of the disparity between the speed of the X.25 link and the ISDN call link. The UPRATE converts to an absolute utilisation of 12.8 kbps (80% of 16 kbps), while the DOWNRATE (when a single ISDN B channel is in use) converts to an absolute utilisation of 16 kbps (20% of 80 kbps). This means that a steady offered load of, say, 14 kbps, will overload the X.25 call on its own and cause a B channel to be added. At this point, however, the offered load will be below DOWNRATE, and the B channel call will be dropped (after a minute). This oscillating pattern of a call being brought up, then dropped, will continue as long as the offered load remains in the band 12.8–16 kbps.

A better set of utilisation parameters might be UPRATE=90% (absolute value of 14.4 kbps) and DOWNRATE=15% (absolute value of 12 kbps). For the transition from the X.25 call plus one B channel to the X.25 call plus two B channels, the absolute values become 72 kbps and 21.6 kbps respectively, which is also a stable configuration. To set these parameters, enter the command:

```
SET PPP=0 UPRATE=90 UPTIME=20 DOWNRATE=15 DOWNTIME=60
```

The UPTIME and DOWNTIME parameters will be set according to how responsive the user requires the router to be to changes in offered load.

Another option altogether is to allow the remote end (in this example, the ISP), to make the decisions on bandwidth allocation. This involves setting the bandwidth parameters to values that ensure that the router will never bring up or take down calls based on them, using the command:

```
SET PPP=0 UPRATE=100 UPTIME=1000000 DOWNRATE=0
DOWNTIME=1000000
```

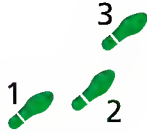
7. Configure IP to use the PPP interface.

The final step is to configure the IP interface which uses the PPP interface:

```
ADD IP INT=PPP0 IP=202.36.163.55 MASK=255.255.255.0
```

Data Over Voice

Some ISDN service providers charge a premium for data calls, compared to voice calls. However, the premium can be avoided by faking a voice call and then sending data over the voice call. Both ends of the link—the device making the call and the device receiving the call—need to be configured correctly to perform this stunt, because if anything special appears in the call setup message the ISDN service provider would be able to detect this and still charge the premium.



To configure data over voice (DOV):

1. Configure the calling router to make voice calls.

The calling router must be configured to use the voice bearer capability when making a DOV call, using either of the following commands:

```
ADD ISDN CALL=call-name DOV={ON|OFF|YES|NO|TRUE|FALSE}
[other-call-options...]
```

```
SET ISDN CALL=call-name DOV={ON|OFF|YES|NO|TRUE|FALSE}
[other-call-options...]
```

The DOV parameter specifies whether or not the outgoing call setup message has data bearer capability or voice bearer capability selected. If DOV is set to ON, voice bearer capability is specified and the ISDN service will treat the call as a voice call. If DOV is set to OFF, data bearer capability is specified and the ISDN service will treat the call as a data call.

2. Configure the answering router to recognise the DOV calls.

The answering router must be configured to recognise a DOV call, by defining a special ISDN number for DOV calls on the Q931 interface:

```
SET q931=interface DOVNUMBER=number
```

The DOVNUMBER parameter specifies an ISDN number for the interface. If a call is received on this interface with a voice bearer capability and a called number matching the value specified for DOVNUMBER, the call is treated as a data call, not a voice call.



The number specified for DOVNUMBER must be a valid ISDN number supplied by the ISDN service provider.

Configuration Examples

The following examples illustrate the steps required to configure ISDN for a range of network functions, from a basic ISDN configuration through to more advanced functionality.

A Basic ISDN Setup

This example illustrates the steps required to configure ISDN calls between two routers as in Figure 4-2 on page 4-33 and Table 4-17 on page 4-33.

Figure 4-2: Example configuration for a basic ISDN network.

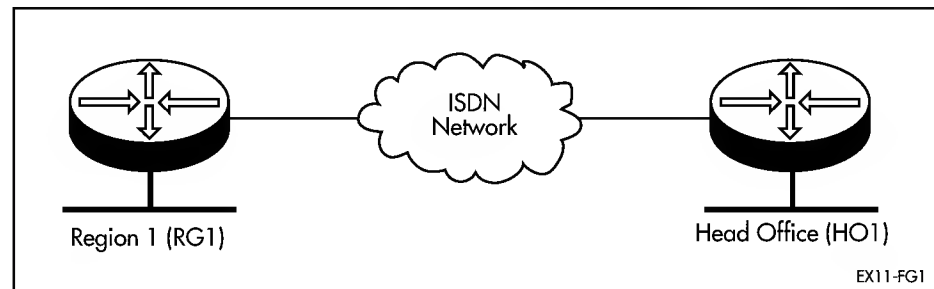
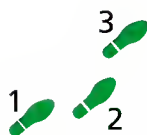


Table 4-17: Example configuration parameters for a basic ISDN network.

Site	Region 1	Head Office
Router Name	RG1	HO1
ISDN Number	1234567	9876543
IP Address for PPP0	192.168.35.114	192.168.35.113
IP Address for Eth0	192.168.35.110	192.168.35.45
Subnet Mask	255.255.255.240	255.255.255.240



To configure a basic ISDN:

ISDN on the router requires minimal user configuration, other than selecting a profile, and creating and enabling calls. The lower layers of the ISDN protocol stack (BRI, LAPD and Q.931) are automatically configured by the SET SYSTEM TERRITORY command on page 1-56 of *Chapter 1, Operation*. Most of the commands associated with these layers are for testing purposes and **should not be used during normal operation** as they may interfere with the functioning of the router.



The factory default hardware and software settings described here are correct for European Union (EU) countries. For other countries, contact your distributor or reseller for details of local requirements.

1. Check the BRI hardware configuration.

Routers and expansion boards with BRI hardware are shipped with the operation mode jumpers set to TE mode and the termination jumpers removed, which are the appropriate settings for normal operation. If the BRI hardware is to be operated as an NT or connected in a point-to-point configuration where termination resistors are not already provided in the

building wiring, these jumpers will need to be changed. Contact your distributor or reseller for details of how to set the jumpers.

The commands:

```
SHOW BRI STATE
SHOW BRI CONFIGURATION
```

can be used to display the state of the BRI interface and the modules that have attached to the BRI interface. No other user configuration is required.

If the interface is to be used to connect to a non-standard ISDN service, the SET BRI command on page 4-73 may be required to alter the mode of operation of the interface:

```
SET BRI=instance [ACTIVATION={NORMAL|ALWAYS}]
[ISDNSLOTS=slot-list] [MODE={ISDN|TDM|MIXED}]
[TDMSLOTS=slot-list]
```

In this case the slots to be used for ISDN calls and TDM (Time Division Multiplexing) groups may need to be defined. See *Chapter 11, Time Division Multiplexing (TDM)* for a detailed description of how to configure TDM groups. Contact your distributor or reseller for assistance with the configuration of the interface.



The MODE parameter of the SET BRI command affects the way the router behaves when connected to a network to the extent that, if configured inappropriately for the network to which it is connected, it may not conform to the national standards applying to that network. Therefore care must be taken when using this command. Please seek the advice of your distributor or ISDN service provider when changing the mode of operation from the default, which is the correct mode for connecting to a standard ISDN network.



Semipermanent connections are not available in the USA and the router will not permit the MODE of a BRI U interface to be set TDM or MIXED or the ACTIVATION mode set to ALWAYS.

2. Check the LAPD configuration.

The LAPD module is automatically configured when the router boots and does not require any user configuration. The LAPD parameters are specified by the LAPD standard and should not be changed without careful consideration. Contact your supplier before using the SET LAPD command on page 4-82.

The command:

```
SHOW LAPD
```

can be used to display the state of the LAPD interface and each DLC.



There is no LAPD entity associated with an interface set to TDM mode.

3. Select a Q.931 profile and set other Q.931 parameters.

The PROFILE parameter determines which network is running on the interface. The profile selected must match the characteristics of the ISDN network to which the router is to be connected (Table 4-18 on page 4-35). The profile is set automatically whenever the router territory is changed by

the SET SYSTEM TERRITORY command on page 1-56 of *Chapter 1, Operation*. The default territory is 'Europe' which sets the profile to ETSI.

Table 4-18: Q.931 Profiles.

Profile Name	Country
5ESS	USA and Canada
AUS	Australia
CHINA	China
DMS-100	USA and Canada
ETSI	European Union countries (ETSI specification)
JAPAN	Japan
KOREA	Korea
NI1	USA and Canada
NZ	New Zealand

To select the Q.931 profile to be used on the ISDN interface, or to override the default set by the SET SYSTEM TERRITORY command on page 1-56 of *Chapter 1, Operation*, use the command:

```
SET Q931=interface PROFILE=profile
```

For example, to set BRI port 0 to use the New Zealand profile, use the command:

```
SET Q931=0 PROFILE=NZB
```



Failure to select the correct profile will invalidate the approval of this product with respect to the applicable national standards for the country in which the product is used. If you are not sure about which profile to use, contact your distributor or ISDN service provider.



There is no Q.931 profile associated with an interface set to TDM mode.

The router's own ISDN numbers and subaddresses may be set with the command:

```
SET Q931=interface NUM1=number NUM2=number SUB1=subaddress  
SUB2=subaddress
```

The numbers and subaddresses must be set when the router is attached to a BRI S/T bus with other TEs, or when SPIDs are being used in this configuration. Two numbers and subaddresses may be defined, although both numbers will only be required when two SPIDs are defined. See below for a description of SPIDs and how they interact with the ISDN numbers.

If the router is the only TE on the bus, all incoming calls will be for the router so the router does not need its own ISDN number. If more than one TE exists on the bus, the incoming SETUP message is sent to all of them, and the called number (and optionally, subaddress) in the SETUP message must be matched with the TE's number before it may reply to the call. The number entered should be the number as supplied by the carrier, without STD access codes or area codes. The incoming number and the router's number will be compared from the right-hand end and only as far as the

shortest of the two numbers. The subaddress specified must not conflict with the subaddresses of other TEs on the bus.

In some networks the router will have to be configured with one or two *Service Profile Identifiers* (SPIDs). A SPID is used to identify the router to the network, and must be correctly configured before calls can be made from the router. The ISDN service provider supplies the SPID(s) for the interface, which are entered with the command:

```
SET Q931=interface SPID1=spid [SPID2=spid]
```

Entry of SPID values is usually tied to entry of ISDN numbers. If two SPIDs are defined, two numbers will have to be defined and the numbers must match the SPIDs. That is, NUM1 must match SPID1 and NUM2 must match SPID2. In most circumstances, the NUM parameter will be a substring of the SPID parameter. When two SPIDs are defined, the router will create two DLCs in the LAPD module, one for each SPID. Calls will be presented to the router on both DLCs, and the router determines, on the basis of the called number, the DLC on which to accept the call. When making outgoing calls, the router will select one of the DLCs. If one DLC already has a call active, the router will select the other DLC.

The SPID facility is only available when the Q.931 profile is one of the Basic Rate profiles NI1, 5ESS, DMS-100 or AUS (Australian Basic Rate). SPIDs are required for the NI1, 5ESS and DMS-100 profiles. The AUS profile will use SPIDs if SPIDs are defined manually. SPIDs are not required for Primary Rate interfaces.

The command:

```
SET Q931 timer=value
```

may be used to set the timeout values for the Q.931 timers (T301, T302, T303, T304, T305, T308, T309, T310, T313, T314, T316, T317, T318, T319, T321 and T322). However, the default values should be adequate for most situations. Contact your distributor or ISDN service provider before making any changes to the Q.931 timers.

The command:

```
SHOW Q931
```

can be used to display the Q.931 profile, the router's numbers and SPIDs, and timer values.

4. Configure ISDN calls.

This step is the only step that actually has to be carried out in order to run ISDN on the router. An ISDN call definition must be created on any two routers that are to communicate with ISDN, using the command:

```
ADD ISDN CALL=name NUMBER=number PRECEDENCE={IN|OUT}
options...
```

As an example, Region 1 is to be connected to Head Office via ISDN. The ISDN number of the Region 1 router is 1234567. The ISDN number of the Head Office router is 9876543 (Figure 4-2 on page 4-33). The ISDN network being used allows the passage of called party subaddress, but CLI is not allowed because of privacy issues and user-user data can only be sent as a subscription option and the facility is not free.

Before a call can be made from one office to the other, call definitions must be created on both routers. In this example, the called party subaddress IE will be used to carry connection information, and PPP interfaces will be created explicitly to use the ISDN calls. Either end will be allowed to initiate the call, but the call from Region 1 will have precedence.

On the Head Office router, create a call to the Region 1 router:

```
ADD ISDN CALL=Region1 OUTSUB=LOCAL SEARCHSUB=LOCAL
NUMBER=1234567 PREC=IN
```

On the Region 1 router, create a call to the Head Office router:

```
ADD ISDN CALL=Region1 OUTSUB=LOCAL SEARCHSUB=LOCAL
NUMBER=9876543 PREC=OUT
```

Note that each call has the same name, and that this name is passed via the *called subaddress Information Element (IE)* to provide identification for the remote end of the link. Each router will search for this call using the called subaddress IE.



The BT implementation of the ETSI specification for European Union countries effectively limits the call name length to 5 characters, for interoperation with other national ISDN services.

The precedence on each call is set to ensure that in the event of a call collision (the same call being made and answered at the same time), the call from Region 1 to Head Office is completed and the reverse call cleared. The direction of precedence is not important, but it is essential the precedence is set to IN at one end of the call and OUT at the other end of the call.

Note that the number entered is the exact sequence required to reach the remote router from the local router, including STD access codes and area codes. Note that the number can contain only decimal digits and that hyphens and other characters will result in an error.

Check that the ISDN calls have been successfully added with the command:

```
SHOW ISDN CALL
```

which for the router at Head Office in the example, will produce a display like that in Figure 4-3 on page 4-37.

Figure 4-3: Example output from the SHOW ISDN CALL command for Head Office.

ISDN call details				
Name	Number	Remote call	State	Precedence
Region1	1234567	-	IN & OUT	IN



The remote call has not been specified for the ISDN call. This is a change from previous versions of the call control software, which required that a remote name be specified. The extra control over the contents of the outgoing SETUP message and how the incoming SETUP message is used in searching for calls means that calls may now be configured in this simpler fashion.

5. Create PPP interfaces to use the ISDN calls.

PPP will be used on the ISDN call just defined. PPP provides the link layer protocol and enables multiple network and transport layer protocols (such as IP) to be carried over the same ISDN link. This is the first PPP instance we will define, so we will number it PPP0. We don't wish to alter any of the

default PPP configuration options for this example, but this will be dealt with in later examples.

On the Head Office router, create PPP0 to use the ISDN call Region1:

```
CREATE PPP=0 OVER=ISDN-Region1
```

On the Region 1 router, create PPP0 to use the ISDN call Region1:

```
CREATE PPP=0 OVER=ISDN-Region1
```

Setting up these PPP instances will cause the ISDN calls to be activated. If the routers are connected to the ISDN at this stage, the call will be connected and the PPP link will be in the OPENED state.

6. Configure routing modules to use the PPP interfaces.

IP will be run over the PPP instance just defined. The IP addresses are given in Table 4-17 on page 4-33. Since the Region 1 router is a stub router, we will reduce use of the ISDN link by setting up static routes at both ends. This means that routing protocol traffic will not flow on the link.

Configure IP at the Head Office router:

```
ENABLE IP
ADD IP INT=ppp0 IP=192.168.35.113 MASK=255.255.255.240
ADD IP ROUTE=192.168.35.96 INT=ppp0 NEXT=192.168.35.114
MET=2
```

Configure IP at the Region 1 router:

```
ENABLE IP
ADD IP INT=ppp0 IP=192.168.35.114 MASK=255.255.255.240
ADD IP ROUTE=0.0.0.0 INT=ppp0 NEXT=192.168.35.113 MET=7
```

7. Test the configuration.

At this stage the ISDN call should be connected and PPP should be open at both the link level and for IP. The configuration should be checked on each router, using the commands:

```
SHOW ISDN CALL
SHOW PPP
SHOW IP INTERFACE
SHOW IP ROUTE
```

The expected output is shown in Figure 4-4 on page 4-39 for the Head Office router, and in Figure 4-5 on page 4-40 for the Region 1 router.

Figure 4-4: Example commands and output to test the configuration of the central site router in a basic ISDN network.

ISDN call details

Name	Number	Remote call	State	Precedence
Region1	1234567	-	IN & OUT	IN

ISDN active calls

Index	Name	Interface	User	State	Prec
0	Region1	BRI0	03-00	ON	Yes

Name	Enabled	ifIndex	Over	CP	State
ppp0	YES	04		IP	OPENED
			ISDN-Region1	PPP	OPENED

Interface	Type	IP Address	Bc	Fr	PArp	Filt	RIP Met.	SAMode	IPSc
Pri. Filt	Pol.Filt	Network Mask	MTU	VJC	GRE	OSPF Met.	DBcast	Mul.	
LOCAL	-	Not Set	-	n	-	---	-	-	---
eth0	Static	192.168.35.45	1	n	On	---	01	Pass	---
		255.255.255.240	1500	-		---	0000000001	None	---
ppp0	Static	192.168.35.113	1	n	-	---	01	Pass	---
		255.255.255.240	1500	Off		---	0000000001	None	---

IP Routes

Destination	Mask	NextHop	Interface	Age
DLCI/Circ.	Type	Policy Protocol	Metrics	Preference
0.0.0.0	0.0.0.0	192.168.35.46	eth0	756
-	remote 0	rip	6	0
192.168.35.32	255.255.255.240	0.0.0.0	eth0	780
-	direct 0	static	1	0
192.168.35.96	255.255.255.240	192.168.35.114	ppp0	715
-	direct 0	static	2	0
192.168.35.112	255.255.255.240	0.0.0.0	ppp0	780
-	direct 0	static	1	0

Figure 4-5: Example commands and output to test the configuration of the regional site router in a basic ISDN network.

```

ISDN call details
Name          Number          Remote call    State    Precedence
-----
Region1       9876543          -             IN & OUT  OUT
-----

ISDN active calls
Index  Name          Interface    User    State  Prec
-----
0      Region1       BRI0        03-00   ON     Yes
-----

Name          Enabled  ifIndex  Over          CP          State
-----
ppp0          YES      04      ISDN-Region1  IP          OPENED
              PPP          OPENED
-----

Interface      Type      IP Address      Bc Fr PArp  Filt  RIP Met.  SAMode  IPSc
Pri. Filt      Pol.Filt Network Mask  MTU  VJC  GRE  OSPF Met.  DBcast  Mul.
-----
LOCAL          -         Not Set         -  n  -    -    -         -      -
---           ---         -         -  -  -    -    -         -      -
eth0           Static    192.168.35.110  1  n  On   -    01         Pass   --
---           ---         255.255.255.240 1500 -    -    00000000001 None  ---
ppp0           Static    192.168.35.114  1  n  -    -    01         Pass   --
---           ---         255.255.255.240 1500 Off  -    00000000001 None  ---
-----

IP Routes
-----
Destination    Mask      NextHop      Interface      Age
DLCI/Circ.     Type      Policy      Protocol      Metrics      Preference
-----
0.0.0.0         0.0.0.0    192.168.35.113 ppp0           697
-              direct    0           static         7            0
192.168.35.0    255.255.255.0 192.168.35.113 ppp0           708
-              direct    0           static         2            0
192.168.35.96   255.255.255.240 0.0.0.0      eth0           726
-              direct    0           static         1            0
192.168.35.112  255.255.255.240 0.0.0.0      ppp0           726
-              direct    0           static         1            0
-----

```

Refining the ISDN Setup

This example builds on the previous example by adding some additional ISDN functionality, including call back facility, minimum call length and call tenacity.

Call Back Facility

The call back facility enables a router connected to an ISDN service to request a remote router to initiate a call to the local router—to “call back”.

Tariffs for ISDN calls vary from country to country, and the cost of a call is determined by the tariffs applying in the country of origin. If a organisation's network spans more than one country, it may be cheaper to make calls in one direction than in the other direction. The call back facility provides the ideal mechanism to manage the cost of international ISDN calls.

The call back facility is enabled with the CALLBACK parameter of the ADD ISDN CALL command on page 4-44 and the SET ISDN CALL command on page 4-75:

```
ADD ISDN CALL=name NUMBER=number PREC={IN|OUT} CALLBACK={ON|
OFF|YES|NO|TRUE|FALSE}
SET ISDN CALL=name CALLBACK={ON|OFF|YES|NO|TRUE|FALSE}
```

The CALLBACK parameter of an incoming call determines whether or not the call is answered (CALLBACK=OFF), or the call is refused and then a call is made to the originator (CALLBACK=ON).

For example, assume that the Head Office router (HO1) and Region 1 router (RG1) are in different tariff zones, and that the tariffs applicable to calls made by the Region 1 router are lower. The call back facility can be enabled using the following command on the Region 1 router:

```
SET ISDN CALL=Region1 CALLBACK=ON
```



If the callback facility is required for specific calls (the normal case), it is necessary to configure the OUTCLI, OUTSUB and OUTUSER parameters in the call definition on the calling router, and the SEARCHCLI, SEARCHSUB and SEARCHUSER parameters in the call definition on the receiving router, to ensure that incoming calls on the receiving router are matched to the call definition with the correct callback.

Minimum Call Length

Some tariff regimes include a base charge for a minimum call length, for example one minute. Calls with a duration of less than the minimum call length will be charged for the minimum call length. In applications where ISDN is used to provide dial-on-demand facilities to other routing protocols (e.g. IP) it may be advantageous for a call, once made, to be kept active for the minimum call length so that additional protocol exchanges can be “piggybacked” on to the call.

A minimum call length can be set with the HOLDUP parameter of the ADD ISDN CALL command on page 4-44 and the SET ISDN CALL command on page 4-75. For example, to set a minimum call length of one minute for calls from the Region 1 router to the Head Office router, on both routers use the command:

```
SET ISDN CALL=Region1 HOLDUP=60
```

Call Tenacity

Call tenacity refers to the concept of keeping an ISDN link as active as possible. When a call fails it is retried until the call is reactivated or for a specified number of attempts have been made. Retries are organised into retry groups. An ISDN call can be assigned one or more retry groups. The number of retries in a group, the number of retry groups, the time interval between retries in a group and the time interval between retry groups can be specified.

A retry regime is established with the RN1, RN2, RT1 and RT2 parameters of the ADD ISDN CALL command on page 4-44 and the SET ISDN CALL command on page 4-75. For example, to set up retry regime to make five retries in the first minute after a call fails and a further five retries 300 seconds later, use the command:

```
SET ISDN CALL=HeadOffice RN1=5 RN2=1 RT1=12 RT2=300
```

Command Reference

This section describes the commands available on the router to configure and manage ISDN.

See “Conventions” on page xxxv of *Preface* in the front of this manual for details of the conventions used to describe command syntax. See *Appendix A, Messages* for a complete list of messages and their meanings.

ACTIVATE ISDN CALL

Syntax `ACTIVATE ISDN CALL=name`

where:

- *name* is an ISDN call name, 1 to 15 characters in length. Valid characters are letters (a–z, A–Z), decimal digits (0–9) and underscore (“_”). It is case-insensitive.

Description This command activates an ISDN call, causing an outgoing ISDN call to be made, using the specified call definition. The call, if its user parameter is ATTACH, must have an attached user module for the call to be made.

Examples To activate the ISDN call “region1”, use the command:

```
ACTIVATE ISDN CALL=region1
```

See Also ADD ISDN CALL
 DEACTIVATE ISDN CALL
 DELETE ISDN CALL
 DISABLE ISDN CALL
 ENABLE ISDN CALL
 SHOW ISDN CALL

ACTIVATE Q931 ASPID

Syntax `ACTIVATE Q931=interface ASPID`

where:

- *interface* is a slotted interface name or number. Interface names are formed by concatenating an interface type and instance (e.g. BRI0 or PRI1). Interface numbers are the decimal index of the slotted interface (0, 1, 2...).

Description This command reinitiates the auto-SPID process. Existing auto-SPID information is deleted, and the auto-SPID process is initiated with the transmission of an INFORMATION message containing the universal SPID ("010101010101") to the network.

If the router has already successfully initialised with a manual or generic SPID, and the auto-SPID process is reinitiated with this command, a failure of the auto-SPID process will result in the manual or generic SPID being retried.

This command can be used in a number of circumstances, for example when the router is moved to a different ISDN interface or the SPID information on the switch changes to allow a new service on the interface.

Examples To manually activate the auto-SPID process for the Basic Rate interface BRI0, use the command:

```
ACTIVATE Q931=bri0 ASPID
```

See Also SHOW Q931 SPID

ACTIVATE Q931 MESSAGE

Syntax `ACTIVATE Q931=interface MESSAGE=message [DLC=dlc-index]`

where:

- *interface* is a slotted interface name or number. Interface names are formed by concatenating an interface type and instance (e.g. BRI0 or PRI1). Interface numbers are the decimal index of the slotted interface (0, 1, 2...).
- *message* is a sequence of hexadecimal digits, each pair of which specifies a single octet in the message. There must be an even number of hexadecimal digits.
- *dlc-index* is the index of a valid DLC on the Q.931 interface, and is one of 1 or 2.

Description This command creates and transmits a message to Q.931, as if it had been received on the Q.931 interface specified.



This command is for debugging only. Use of this command in normal operation will probably result in strange and unexpected behaviour in the Q.931 operations of the router.

The MESSAGE parameter specifies a sequence of octets which form the message. Since each octet requires 2 hexadecimal digits, an even number of hexadecimal digits must be specified. The first octet in the message is the first octet of the Q.931 message, which is always the Q.931 protocol discriminator.

The DLC parameter specifies the DLC on which the message is to be received. The valid DLCs are 1 and 2. If this parameter is not specified, DLC 1 will be used.

Examples To send the router an ALERTING message which tests the reception of an unexpected message, use the command:

```
ACTIVATE Q931=0 MESSAGE="08018301"
```

See Also ENABLE Q931 DEBUG

ADD ISDN CALL

Syntax ADD ISDN CALL=*name* NUMBER=*number* PRECEDENCE={IN|OUT} [ALTNUMBER=*number*] [BUMPDELAY=0..100] [CALLBACK={ON|OFF|YES|NO|TRUE|FALSE}] [CALLINGNUMBER=*number*] [CALLINGSUBADDRESS=*calling-subaddress*] [CBDELAY=0..100] [CHECKCLI={OFF|PRESENT|REQUIRED}] [CHECKSUB={OFF|LOCAL|REMOTE}] [CHECKUSER={OFF|LOCAL|REMOTE}] [CLILIST=0..99] [DIRECTION={IN|OUT|BOTH}] [DOV={ON|OFF|YES|NO|TRUE|FALSE}] [HOLDUP=0..7200] [INANY={ON|OFF|YES|NO|TRUE|FALSE}] [INTPREF={NONE|*interface*}] [INTREQ={NONE|*interface*}] [KEEPUUP={ON|OFF|YES|NO|TRUE|FALSE}] [LOGIN={ALL|NONE|CHAP|PAP-RADIUS|PAP-TACACS|USER}] [OUTCLI={OFF|CALLING|INTERFACE|NONNUMBER}] [OUTSUB={OFF|LOCAL|REMOTE}] [OUTUSER={OFF|LOCAL|REMOTE}] [PASSWORD={NONE|CLI|CALLED SUB|NAME|USER}] [PPPTEMPLATE=*template*] [PRIORITY=0..99] [RATE={56K|64K}] [REMO TECALL=*name|remote-number*] [RN1=0..10] [RN2=0..5] [RT1=5..120] [RT2=300..1200] [SEARCHCLI={ON|OFF|YES|NO|TRUE|FALSE|CALLED|0..99}] [SEARCHSUB={OFF|LOCAL|REMOTE}] [SEARCHUSER={OFF|LOCAL|REMOTE}] [SUBADDRESS=*number*] [USER={ATTACH|PPP}] [USERNAME={NONE|CLI|CALLED SUB|NAME|USER}]

where:

- *name* is an ISDN call name, 1 to 15 characters in length. Valid characters are letters (a-z, A-Z), decimal digits (0-9), hyphen ("-") and underscore ("_"). It is case-insensitive.
- *number* is an ISDN phone number, 1 to 31 characters in length. Valid characters are decimal digits (0-9).
- *calling-subaddress* is a character string, 1 to 31 characters in length. Valid characters are letters (a-z, A-Z), decimal digits (0-9) and underscore ("_"). It is case-insensitive.
- *interface* is a slotted interface name or number. Interface names are formed by concatenating an interface type and instance (e.g. BRI0 or PRI1). Interface numbers are the decimal index of the slotted interface (0, 1, 2...).
- *template* is a number in the range 0 to 31.

- *remote-number* is a number, 1 to 15 characters in length. Valid characters are decimal digits (0–9).

Description This command creates a new ISDN call definition. The CALL, NUMBER and PRECEDENCE parameters are required. Other parameters will probably be required in order to actually get the call to work.

The CALL parameter uniquely identifies this call in the router. All commands that affect ISDN call definitions must specify the call with this parameter. ISDN call names are case-insensitive. The case of the ISDN call name as entered will be saved, so case can be used to provide readable names. However, any form of the name can be used in subsequent commands, and no two calls may have the same name when case is ignored.

The NUMBER parameter specifies the number called when this call is activated. This is the number that Q.931 uses in the SETUP message passed to the network, so it must include all access and area codes required by the network and be formatted in the way required by the network. Spaces or other characters may not be entered in between the digits of the number.

The PRECEDENCE parameter specifies the direction of precedence for the call in the event of call collision. Call collision occurs when a call is activated at the same time as an incoming call selects the same call. If precedence is IN, the incoming call has precedence and the outgoing call is cleared. If precedence is OUT, the outgoing call has precedence and the incoming call is cleared.

The ALTNUMBER parameter specifies an alternate ISDN number for this call to ring if all retries and retry groups for the main number have failed. The ISDN call retry parameters (RN1, RN2, RT1 and RT2) apply only to the main ISDN number. The alternate number is tried only once. The KEEPU parameter, if set, forces ISDN call control to cycle repeatedly through the main number, all retries and retry groups for the main number, and then the alternate number, until a call succeeds.

The BUMPDELAY parameter specifies the time, in tenths of a second, the router will wait after bumping another call before initiating this call. Call bumping involves clearing a call and using the resulting free B channel for a new call. A delay is programmable with the BUMPDELAY parameter in order to give the network time to clear the bumped call's B channel. The default is 5, that is, 0.5s.

The CALLBACK parameter specifies whether this call, upon being selected by an incoming call, should clear the incoming call and call back or not. The values ON, TRUE and YES are equivalent and mean that the call back will occur. The values OFF, FALSE and NO are equivalent, and mean that the call back will not occur. The default value is NO.

The CALLINGNUMBER parameter may be used in connecting this call to a remote call. Certain options for formatting the outgoing SETUP message allow the calling number to be specified.

The CALLINGSUBADDRESS parameter specifies a calling subaddress to be placed in the outgoing SETUP message. This value will be placed in the outgoing SETUP message only if the OUTCLI parameter is set to CALLING.

The CBDELAY parameter specifies the time, in tenths of a second, the router will wait after clearing a call before initiating a callback for the call. Call back involves clearing a call and using the resulting free B channel for the new call. A delay is programmable with the CBDELAY parameter in order to give the

network time to clear the incoming call's B channel. The default is 41, that is, 4.1s.

The CHECKCLI parameter specifies how this call, if selected, is checked against the CLI IE in the incoming SETUP message. The check, if carried out, consists of verifying that the CLI number appears in the CLI list for this call. The default value of OFF means that no check is carried out. The value PRESENT means that the check is carried out only if the CLI IE is present, and contains calling number digits. The check passes if the CLI IE is not present, or does not contain calling number digits, or is present and contains a matching CLI number. The value REQUIRED means that CLI MUST be present, and must contain calling number digits. The check fails if the CLI IE is not present, or does not contain calling number digits, or does not contain a matching CLI number.

The CHECKSUB parameter specifies whether this call, when selected, should have the called party subaddress IE of the incoming SETUP message checked. The IE may be checked against the call name (parameter set to LOCAL) or the remote call name (parameter set to REMOTE). The default value is OFF, which means that no check is carried out.

The CHECKUSER parameter specifies whether this call, when selected, should have the user-user data IE of the incoming SETUP message checked. The IE may be checked against the call name (parameter set to LOCAL) or the remote call name (parameter set to REMOTE). The default value is OFF, which means that no check is carried out.

The CLILIST parameter specifies the CLI list against which this call is checked if the check CLI parameter is either PRESENT or REQUIRED. The default value is a special value that means that the list is undefined.

The DIRECTION parameter specifies the directions for which the call is enabled. Calls may be enabled both for sending and receiving calls, or for either direction. The default value is BOTH.

The DOV parameter specifies whether or not the outgoing call setup message for this call has data bearer capability or voice bearer capability. If DOV is set to ON, voice bearer capability is specified and the ISDN service will treat the call as a voice call. If DOV is set to OFF, data bearer capability is specified and the ISDN service will treat the call as a data call. The values ON and TRUE are equivalent to YES. The values OFF and FALSE are equivalent to NO. The default is NO. The DOV parameter is used in conjunction with the DOVNUMBER parameter on the SET Q931 command on page 4-84 to configure data over voice (DOV).

The HOLDUP parameter specifies the minimum time, in seconds, that this call should be held up after activation. If the user of the ISDN call requests a deactivation, and the holdup time has not expired, the deactivation will be ignored until the holdup time has expired. The default for this parameter is 0 seconds.

The INANY parameter specifies whether this call may be selected to match any incoming call. The search for calls with INANY set YES follows all other searches. Only one call should have INANY set to YES, since otherwise a predictable response to incoming calls cannot be guaranteed. The default value for this parameter is NO.

The INTREQ parameter specifies which ISDN interface MUST be used for this call, when the call is activated as an outgoing call. If no channel is available on

the required interface, the call will fail. The default for this parameter is NONE, which means no required interface.

The INTPREF parameter specifies which ISDN interface should preferentially be used for this call, if the required interface is not specified. When activating this call, the preferred interface is checked first for a free channel. If no free channel is found, other interfaces may be checked. The default for this parameter is NONE, which means no preferred interface.

The KEEPU parameter determines whether the call should be kept up at all costs or not. The KEEPU parameter for a call is inspected when all retries for the main number have failed and the alternate number (if defined) has also failed, and when the call is cleared for any reason other than explicit clearing by the user module or by manager command. If the KEEPU parameter has the value YES, the call will be reactivated in these circumstances. The values ON and TRUE are equivalent to YES. The values OFF, FALSE and NO are equivalent for turning off the KEEPU parameter. The default value is NO.

The LOGIN parameter specifies which login procedure this call must use when it is activated. If CHAP is specified, the call will be accepted but will create a PPP interface which will authenticate using CHAP. If PAP-RADIUS is specified, the call will be accepted but will create a PPP interface which will authenticate using PAP, and using RADIUS as the means of authenticating the PAP exchange. If PAP-TACACS is specified, the call will be accepted but will create a PPP interface which will authenticate using PAP, and using TACACS as the means of authenticating the PAP exchange. If USER is specified, the User Authentication Database in the router is checked. The default is NONE, which means that no login procedure is required.

The values CHAP, PAP-TACACS and PAP-RADIUS are only used when the ISDN call creates a dynamic PPP interface. Since these parameters can also be set by defining a PPP template with the appropriate authentication parameters, use of these values is for backward compatibility only. The value specified in the LOGIN parameter will override the authentication settings in the PPP template.

The OUTCLI parameter specifies the format of the calling party number IE and calling subaddress IE (also known as CLI) in the outgoing SETUP message created when this call is activated. If OFF is specified, the CLI is not included in the SETUP message. If CALLING is specified, the calling number and calling subaddress values from the ISDN call definition are placed in the SETUP message. If the CALLINGSUBADDRESS parameter is not defined, the calling subaddress IE will not be included in the SETUP message. If INTERFACE is specified, the number and subaddress values from the Q.931 interface (set with the SET Q931 command on page 4-84) are placed in the SETUP message. If the Q.931 interface does not have a subaddress set, the calling subaddress IE will not be included in the SETUP message. If NONUMBER is specified, an empty calling number IE and the calling subaddress from the Q.931 interface (if set) are included in the SETUP message. The ISDN itself can fill in the calling number IE in the SETUP message before sending the message to the remote end. The default is OFF.

The OUTSUB parameter specifies the format of the called party subaddress IE in the outgoing SETUP message created when this call is activated. The default value for this parameter is OFF, which means that the called party subaddress IE is not included in the SETUP. The call name or remote call name may be specified.

The OUTUSER parameter specifies the format of the user-user data IE in the outgoing SETUP message created when this call is activated. The default value for this parameter is OFF, which means that the user-user data IE is not included in the SETUP. The call name or remote call name may be specified.

The PASSWORD parameter specifies the source of the password for login procedures. The default value of NONE means that no password is specified. The values CLI, CALLED SUB and USER mean that the password is drawn from, respectively, the CLI, called party subaddress and user-user data IE in the incoming SETUP message. The value of NAME means that the call name is used as the password.

The PPPTEMPLATE parameter specifies the PPP template to use when creating a dynamic PPP interface for this call. The specified template must exist. See *“Templates”* on page 3-9 of *Chapter 3, Point-to-Point Protocol (PPP)* for more information about creating PPP templates.

The PRIORITY parameter specifies the priority of this call for use by the call bumping facility. The value of this parameter is a number in the range 0 to 99. The default is 50. Table 4-15 on page 4-25 details how the different priority values affect the bumping of data calls.

The RATE parameter specifies the rate of data transmitted and received on the B channel for this call. The rate can be either 64 kbps (the default value) which is the full bandwidth of the B channel, or 56 kbps, which is specified by ITU-T standard V.110 (rate adaption). The data rate specified by this parameter will be used when this call is used as an outgoing call. When the call is selected as an incoming call, the data rate is determined by the bearer capability in the SETUP message or the rate set for the entire Q.931 interface, as specified by the SET Q931 command on page 4-84.

The REMOTECALL parameter may be used in connecting this call to a remote call. Certain options for formatting the outgoing SETUP message and searching for calls allow the remote call to be specified. This parameter has the same syntax as the CALL parameter except that all numeric entries are allowed, for interoperation with devices that can only send numeric subaddresses.

The REMOTECALL parameter is used to connect this call to a remote call. Some options for formatting the outgoing SETUP message and searching for calls allow the remote call to be specified. This parameter has the same syntax as the CALL parameter.

The RN1 parameter specifies how many times this call will be retried in a single retry group. The default value of 0 means that the call will not be retried in a retry group.

The RN2 parameter specifies how many retry groups this call will have, after the first group. The default value of 0 means that the first group only will be tried.

The RT1 parameter specifies the time in seconds between retries in the same retry group. The default is 30 seconds.

The RT2 parameter specifies the time in seconds between retry groups. The default is 600 seconds.

The SEARCHCLI parameter specifies whether this call may be included in a search based on the CLI IE in the incoming SETUP message. If ON is specified,

the value of the CLI IE in the incoming SETUP message is compared with the called number (NUMBER) parameter of this call definition. The options TRUE, YES and CALLED are synonyms for ON. If OFF is specified, there is no search based on the CLI IE. The options FALSE and NO are synonyms for OFF. If a number is specified it identifies an existing CLI list, and the value of the CLI IE is compared with all numbers in the specified CLI list. The default value is OFF.

The SEARCHSUB parameter specifies whether this call may be included in a search based on the called party subaddress IE in the incoming SETUP message. In such a search, the called party subaddress IE may be compared with the call name (parameter set to LOCAL) or the remote call name (parameter set to REMOTE). The default value is OFF.

The SEARCHUSER parameter specifies whether this call may be included in a search based on the user-user data IE in the incoming SETUP message. In such a search, the user-user data IE may be compared with the call name (parameter set to LOCAL) or the remote call name (parameter set to REMOTE). The default value is OFF.

The SUBADDRESS parameter allows the specification of an entirely numeric subaddress to be placed in the outgoing SETUP message when this call is activated. The subaddress as specified by the OUTSUB parameter has the limitation that it can only be the remote or local call name, which means that entirely numeric subaddresses cannot be specified with this parameter alone. However, in some cases, a numeric subaddress is required to satisfy network requirements when calling a router which shares an S/T bus with other ISDN devices. The default value for this parameter is a null (empty) string. If this parameter has a value, it overrides the OUTSUB parameter when setting the called subaddress IE in the outgoing SETUP message.

The USER parameter specifies how users of ISDN calls use this call. The value ATTACH, the default, means that users must attach to this call before it can be used. The value PPP means that this call is able to create dynamic PPP interfaces when activated. The PPP value is most likely to be used for incoming ISDN calls, which use the user data base to set parameters for the PPP and IP interfaces dynamically created.

The USERNAME parameter specifies the source of the user name for login procedures. The default value of NONE means that no user name is specified. The values CLI, CALLEDSUB and USER mean that the user name is drawn from, respectively, the CLI, called party subaddress and user-user data IE in the incoming SETUP message. The value of NAME means that the call name is used as the user name.

Examples To create a call named "ROHO" to make a call from a Regional Office to the Head Office (number 9876543), with calls to Head Office taking precedence over calls from Head Office, use the command:

```
ADD ISDN CALL=ROHO OUTSUB=LOCAL INANY=TRUE SEARCHSUB=LOCAL  
NUMBER=9876543 PREC=OUT
```

See Also ACTIVATE ISDN CALL
DEACTIVATE ISDN CALL
DELETE ISDN CALL
DISABLE ISDN CALL
ENABLE ISDN CALL
SHOW ISDN CALL

ADD ISDN CLILIST

Syntax `ADD ISDN CLILIST=0..99 NUMBER=number`

where:

- *number* is an ISDN phone number, 1 to 31 characters in length. Valid characters are decimal digits (0–9).

Description This command adds a specified ISDN phone number to a specified CLI list. CLI lists are numbered from 0 to 99 inclusive.

The NUMBER parameter specifies the ISDN number to add to the CLI list. This number is used in comparisons with the number in the CLI information element (IE) in incoming SETUP messages, when the ISDN call selected has options set which required a search of a CLI list. The comparison takes place from the end of the numbers to the beginning, and stops when the shorter number has been checked. For example the number 3432114 in an incoming CLI IE would match CLI list numbers 2114, 3432114 and 033432114.

Examples To add the number (412) 986-0117 to CLI list 1, use the command:

```
ADD ISDN CLILIST=1 NUMBER=4129860117
```

See Also DELETE ISDN CLILIST
SHOW ISDN CLILIST

ADD ISDN DOMAINNAME

Syntax `ADD ISDN DOMAINNAME=domain-name`

where:

- *domain-name* is a domain name.

Description This command defines a domain name to be prepended to a login name for a DNS lookup to determine the IP address to be used for an ISDN call. Only one domain name may be defined.

Examples To specify the domain name "acc.newco.co.nz" for use with DNS lookups, use the command:

```
ADD ISDN DOMAINNAME=acc.newco.co.nz
```

See Also DELETE ISDN DOMAINNAME
SET ISDN DOMAINNAME
SHOW ISDN DOMAINNAME

ADD LAPD TEI

Syntax `ADD LAPD=interface TEI=tei...`

where:

- *interface* is the slotted interface number (0, 1, 2,...).
- *tei* is a TEI value, in the range 0 to 63.

Description This command is used in non-automatic TEI assignment mode to add a TEI to the interface.



This command is not required for normal operation. It should only be used for a BRI interface in non-automatic TEI assignment mode.

Examples To add TEI 32 to LAPD interface 0, use the command:

```
ADD LAPD=0 TEI=32
```

See Also DELETE LAPD TEI
SET LAPD
SHOW LAPD

ADD LAPD XSPID

Syntax ADD LAPD=*interface* XSPID=*spid-index*

where:

- *interface* is the slotted interface number (0, 1, 2,...).
- *spid-index* is a SPID index, 1 or 2.

Description This command is used for packet mode (X.25 on the D channel) operations on Basic Rate interfaces to add a SPID index for the purposes of TEI assignment. This command identifies to LAPD which Q.931 SPID or SPIDs are valid for the packet mode. When LAPD allocates a TEI for packet mode connections, it will assign the same TEI as the Q.931 connection whose SPID index is specified with this command.

The XSPID parameter specifies which of the valid Q.931 SPID indices is being added. The only valid SPID indices are 1 and 2.

The use of the XSPID indices added with this command is overridden if fixed TEIs are defined for packet mode operations using the ADD LAPD XTEI command on page 4-52.

Examples To use SPID 2 for packet mode connections on LAPD interface 0, use the command:

```
ADD LAPD=0 XSPID=2
```

See Also DELETE LAPD XSPID
SHOW LAPD

ADD LAPD XTEI

Syntax `ADD LAPD=interface XTEI=tei`

where:

- *interface* is the slotted interface number (0, 1, 2,...).
- *tei* is a TEI value, in the range 0 to 63.

Description This command is used for packet mode operations on Basic Rate interfaces to add a fixed TEI that can be used explicitly for packet mode operations. This command can be used regardless of whether the LAPD interface has been set for automatic or non-automatic TEI operation.

This command should only be used where packet mode operations must use a fixed TEI. This fact should be made clear to the user when the packet mode service is ordered from the ISDN network supplier. Any TEI value can be used, but care must be taken that values are unique over all terminal equipment on the S/T bus.

Examples To assign a fixed TEI of 56 for packet mode connections on LAPD interface 1, use the command:

```
ADD LAPD=1 XTEI=56
```

See Also DELETE LAPD XTEI
SHOW LAPD

DEACTIVATE ISDN CALL

Syntax `DEACTIVATE ISDN CALL={acnum|name}`

where:

- *acnum* is the index of an active ISDN call.
- *name* is an ISDN call name, 1 to 15 characters in length. Valid characters are letters (a–z, A–Z), decimal digits (0–9) and underscore (“_”). It is case-insensitive.

Description This command deactivates either a particular ISDN active call, or all active calls tied to a particular call definition. If an active call index is specified, only that call is deactivated. If a call name is given, all calls for that call definition are deactivated.

The SHOW ISDN CALL command on page 4-103 may be used to determine the index of active calls.

Examples To deactivate the ISDN call “Region1”, use the command:

```
DEACTIVATE ISDN CALL="Region1"
```

See Also ACTIVATE ISDN CALL
 ADD ISDN CALL
 DELETE ISDN CALL
 DISABLE ISDN CALL
 ENABLE ISDN CALL
 SHOW ISDN CALL

DELETE ISDN CALL

Syntax DELETE ISDN CALL=*name*

where:

- *name* is an ISDN call name, 1 to 15 characters in length. Valid characters are letters (a–z, A–Z), decimal digits (0–9) and underscore (“_”). It is case-insensitive.

Description This command deletes an ISDN call definition. The call definition will not be deleted if there are active calls using this definition, or if there are users (such as PPP) attached to the call definition.

Examples To delete ISDN call “ROHO, use the command:

```
DELETE ISDN CALL=ROHO
```

See Also ACTIVATE ISDN CALL
 ADD ISDN CALL
 DEACTIVATE ISDN CALL
 DISABLE ISDN CALL
 ENABLE ISDN CALL
 SHOW ISDN CALL

DELETE ISDN CLILIST

Syntax DELETE ISDN CLILIST=0..99 NUMBER=*number*

where:

- *number* is an ISDN phone number, 1 to 31 characters in length. Valid characters are decimal digits (0–9).

Description This command removes a specified ISDN phone number from a specified CLI list. CLI lists are numbered from 0 to 99 inclusive.

The NUMBER parameter specifies the ISDN number to remove from the CLI list. The number must exactly match an existing number in the CLI list.

Examples To delete the number (412) 986-0117 from CLI list 1, use the command:

```
DELETE ISDN CLILIST=1 NUMBER=4129860117
```

See Also ADD ISDN CLILIST
 SHOW ISDN CLILIST

DELETE ISDN DOMAINNAME

Syntax DELETE ISDN DOMAINNAME [=domain-name]

where:

- domain-name is a domain name.

Description This command deletes the ISDN domain name definition used for DNS lookups. Only one domain name may be defined.

Examples To delete the ISDN domain name, use the command:

```
DELETE ISDN DOMAINNAME
```

See Also ADD ISDN DOMAINNAME
SET ISDN DOMAINNAME

DELETE LAPD TEI

Syntax DELETE LAPD=interface TEI=tei

where:

- interface is the slotted interface number (0, 1, 2,...).
- tei is a TEI value, in the range 0 to 63.

Description This command is used in non-automatic TEI assignment mode to delete a TEI from the slotted interface. Any connections using the TEI are released. Any calls which use the DLC will be halted.



This command is not required for normal operation. It should only be used for a BRI interface in non-automatic TEI assignment mode.



This command may also be used in automatic TEI mode, but this is not recommended as it may confuse the ISDN switch.

Examples To delete TEI 32 from LAPD interface 0, use the command:

```
DELETE LAPD=0 TEI=32
```

See Also ADD LAPD TEI
SET LAPD
SHOW LAPD

DELETE LAPD XSPID

Syntax DELETE LAPD=interface XSPID=spid-index

where:

- *interface* is the slotted interface number (0, 1, 2,...).
- *spid-index* is a SPID index, 1 or 2.

Description This command is used for packet mode (X.25 on the D channel) operations on Basic Rate interfaces to delete a SPID index for the purposes of TEI assignment. This command identifies to LAPD which Q.931 SPID or SPIDs are valid for the packet mode. When LAPD allocates a TEI for packet mode connections, it will assign the same TEI as the Q.931 connection whose SPID index is specified with this command.

The XSPID parameter specifies which of the valid Q.931 SPID indices is being added. The only valid SPID indices are 1 and 2.

The use of the XSPID indices added with this command is overridden if fixed TEIs are defined for packet mode operations using the ADD LAPD XTEI command on page 4-52.

Examples To remove SPID 2 as the SPID for packet mode connections on LAPD interface 0, use the command:

```
DELETE LAPD=0 XSPID=2
```

See Also ADD LAPD XSPID
SHOW LAPD

DELETE LAPD XTEI

Syntax DELETE LAPD=*interface* XTEI=*tei*

where:

- *interface* is the slotted interface number (0, 1, 2,...).
- *tei* is a TEI value, in the range 0 to 63.

Description This command is used for packet mode operations on Basic Rate interfaces to delete a fixed TEI for packet mode operations. To change the TEI used for packet mode operations, the existing TEI must be deleted and the new TEI added.

Examples To delete TEI 56 as the fixed TEI for packet mode connections on LAPD interface 1, use the command:

```
DELETE LAPD=1 XTEI=56
```

See Also ADD LAPD XTEI
SHOW LAPD

DISABLE BRI CTEST

Syntax DISABLE BRI=*instance* CTEST

where:

- *instance* is the number of the BRI interface.

Description This command disables the currently running conformance test on the BRI interface. Only one conformance test may be running at one time (Table 4-19 on page 4-60).



This command is required for conformance testing only, and should not be used for normal operation of the BRI interface.

Examples To disable the conformance test currently running on BRI interface 1, use the command:

```
DISABLE BRI=1 CTEST
```

See Also ENABLE BRI CTEST
DISABLE BRI TEST
ENABLE BRI TEST
SHOW BRI CTEST
SHOW BRI TEST

DISABLE BRI DEBUG

Syntax DISABLE BRI [=*instance*] DEBUG [= {ERRORS | INDICATIONS | STATES | EVENTS | ALL}]

where:

- *instance* is the number of the BRI interface.

Description This command disables the specified debug option on the BRI interface. If an interface is not specified, the debug option is disabled on all BRI interfaces. If a debug option is not specified, all debug options currently enabled on the interface are disabled (Table 4-20 on page 4-62). Only a single debug option can be disabled on each invocation. Successive commands can be used to disable any combination of debug options.

Examples To enable the STATE and EVENT debug options on all BRI interfaces, use the command sequence:

```
DISABLE BRI DEBUG=ALL
ENABLE BRI DEBUG=STATES
ENABLE BRI DEBUG=EVENTS
```

See Also ENABLE BRI DEBUG
SHOW BRI DEBUG

DISABLE BRI TEST

Syntax `DISABLE BRI=instance TEST[=test-number]`

where:

- *instance* is the number of the BRI interface.
- *test-number* is the number of the test to be disabled.

Description This command disables the specified test on the BRI interface. If a test is not specified, all tests currently running on the interface are disabled. Only a single test can be disabled on each invocation. Successive commands can be used to disable any combination of tests. The tests available depend on whether the BRI interface uses a MC145474 S/T interface controller (Table 4-21 on page 4-62), a PSB2186 S/T interface controller (Table 4-22 on page 4-63), a PEB2091 U interface controller (Table 4-23 on page 4-63) or a MC145572 U interface controller (Table 4-24 on page 4-64).



This command is required for testing only, and should not be used for normal operation of the BRI interface.

Examples To enable tests 8 and 9 on interface BRI0, use the commands:

```
DISABLE BRI=0 TEST
ENABLE BRI=0 TEST=8
ENABLE BRI=0 TEST=9
```

See Also DISABLE BRI CTEST
ENABLE BRI CTEST
ENABLE BRI TEST
SHOW BRI CTEST
SHOW BRI TEST

DISABLE ISDN CALL

Syntax `DISABLE ISDN CALL=name`

where:

- *name* is an ISDN call name, 1 to 15 characters in length. Valid characters are letters (a–z, A–Z), decimal digits (0–9) and underscore (“_”). It is case-insensitive.

Description This command disables an ISDN call definition. Existing active calls for this call are unaffected.

Examples To disable ISDN call “ROHO”, use the command:

```
DISABLE ISDN CALL=ROHO
```

See Also `ACTIVATE ISDN CALL`
`ADD ISDN CALL`
`DEACTIVATE ISDN CALL`
`DELETE ISDN CALL`
`ENABLE ISDN CALL`
`SHOW ISDN CALL`

DISABLE ISDN LOG

Syntax `DISABLE ISDN LOG`

Description This command disables the ISDN call logging facility. The call logging facility records details of events associated with ISDN calls.

An entry is added to the log when a call is initiated. When the log exceeds a predefined maximum length, the oldest entry that is in the CLEARED state is removed from the log. If no entries qualify the log is allowed to grow larger than the maximum defined length. Log messages can be sent to an asynchronous port on the router when the log entry enters the CLEARED state. The maximum length of the log and the port to which messages should be sent can be set with the SET ISDN LOG command on page 4-81.

The forwarding of ISDN log messages to the router’s logging facility is not affected by the status of the ISDN call logging facility.

Examples To disable ISDN call logging, use the command:

```
DISABLE ISDN LOG
```

See Also `DISABLE Q931 DEBUG`
`ENABLE ISDN LOG`
`ENABLE Q931 DEBUG`
`SET ISDN LOG`
`SHOW ISDN LOG`

DISABLE Q931 DEBUG

Syntax `DISABLE Q931=interface DEBUG={MDECODE|MRAW|SDLC|
SINTERFACE|SSPID|SSPIDFILE|STATE|TRACE}`

where:

- *interface* is a slotted interface name or number. Interface names are formed by concatenating an interface type and instance (e.g. BRI0 or PRI1). Interface numbers are the decimal index of the slotted interface (0, 1, 2...).

Description This command disables the Q.931 debug option on the specified slotted interface.

The MDECODE option displays each Q.931 message in a decoded format (Figure 4-6 on page 4-67, Table 4-25 on page 4-68). The message header is decoded, along with the type of each information element (IE) in the message. The octets of each IE are displayed in hexadecimal digit format, and for some IEs, a further decode is provided. If MDECODE debugging is enabled on an interface on which MRAW debugging is already enabled, then the MRAW display will be turned off, so that only the decoded messages will be displayed.

The MRAW debug option displays, for each Q.931 message sent or received on the specified interface, a display of the octets in the message, with no interpretation (Figure 4-7 on page 4-68, Table 4-26 on page 4-68). Each octet is displayed as two hexadecimal digits.

The SDLC option displays all DLC state machine events and state changes, as they occur, for the given interface (Figure 4-8 on page 4-69, Table 4-27 on page 4-69). The DLC state machine controls the activation and deactivation of DLCs on the Q.931 interface.

The SINTERFACE option displays all interface state machine events and state changes, as they occur, for the given interface. The interface state machine controls automatic switch detection as well as other procedures related to bringing an interface to a usable state.

The SSPID option displays all SPID state machine events and state changes, as they occur, for the given interface (Figure 4-9 on page 4-69, Table 4-28 on page 4-69). The SPID state machine controls the initialisation of the DLC via the SPID procedures, as well as auto-SPID detection.

The SSPIDFILE option displays all SPID file state machine events and state changes (Figure 4-10 on page 4-70, Table 4-29 on page 4-70). The SPID file state machine is concerned with controlling which SPID (generic, manual, auto-SPID) is actually used in initialising.

The STATE option displays all call state events and state changes (Figure 4-11 on page 4-70, Table 4-30 on page 4-71). The call state machine takes an ISDN call from initiation through to establishment to disconnection.

The TRACE option displays all subroutine calls within the Q.931 module. This option is not related to a particular interface, so while the interface must be entered as part of the command, subroutine tracing for all interfaces will be enabled.

Examples To disable the display of decoded Q.931 messages sent and received via Q.931 interface 0, use the command:

```
DISABLE Q931=0 DEBUG=MDECODE
```

See Also DISABLE ISDN LOG
ENABLE ISDN LOG
ENABLE Q931 DEBUG
SET ISDN LOG
SHOW ISDN LOG

DISABLE RAPI

Syntax DISABLE RAPI

Description This command disables Remote CAPI (RAPI). All current RAPI sessions are terminated and all existing calls initiated via RAPI are cleared. The router will not accept connections from RAPI clients until RAPI is enabled again. RAPI is disabled by default and must be enabled to support RAPI clients.

Examples To disable remote CAPI, use the command:

```
DISABLE RAPI
```

See Also ENABLE RAPI

ENABLE BRI CTEST

Syntax ENABLE BRI=*instance* CTEST=*test-number*

where:

- *instance* is the number of the BRI interface.
- *test-number* is the number of the conformance test to be enabled.

Description This command enables the specified conformance test on the BRI interface. Only one conformance test may be running at any one time. No other conformance test may be currently running on the interface (Table 4-19 on page 4-60).

Table 4-19: ISDN Basic Rate Interface conformance tests.

Test	Function
1	An activation request is issued to the transceiver which will transmit INFO 1 in an attempt to activate the S/T loop. The status of the test is reset to "no" once the loop activates or when the activate timer times out. This conformance test has no effect if the loop is already activated.
2	Data received by the BRI module for both B channel and the D channel from the S/T loop is retransmitted on the same channel. This corresponds to loopback 4 defined in Appendix I of ITU-T Recommendation I.430.

Table 4-19: ISDN Basic Rate Interface conformance tests. (Continued)

Test	Function
3	HDLC frames containing all zeroes is transmitted continuously on both B channels.
4	High priority HDLC frames containing a fox message are transmitted on the D channel continuously.
5	Low priority HDLC frames containing a fox message are transmitted on the D channel continuously.
6	HDLC frames containing a fox message are transmitted on the B1 channel continuously.
7	HDLC frames containing a fox message are transmitted on the B2 channel continuously.
8	HDLC frames containing bytes with one zero and seven ones are transmitted on the D channel continuously.



This command is required for conformance testing only, and should not be used for normal operation of the BRI interface.

Examples To enable conformance test 8 on BRI interface 1, use the command:

```
ENABLE BRI=1 CTEST=8
```

See Also DISABLE BRI CTEST
DISABLE BRI TEST
ENABLE BRI TEST
SHOW BRI CTEST
SHOW BRI TEST

ENABLE BRI DEBUG

Syntax `ENABLE BRI [=instance] DEBUG [= {ERRORS | INDICATIONS | STATES | EVENTS | ALL}]`

where:

- *instance* is the number of the BRI interface.

Description This command enables the specified debug option on the BRI interface. If an interface is not specified, the debug option is enabled on all BRI interfaces. If a debug option is not specified, all debug options are enabled on the interface(s) (Table 4-20 on page 4-62). Only a single debug option can be enabled on each invocation. Successive commands can be used to enable any desired combination of debug options.

Table 4-20: ISDN Basic Rate Interface debug options.

Category	Meaning
Errors	A BRI software module internal error.
Indications	An indication from the layer 1 state machine to a higher layer or the management layer.
State changes	A change of state for the layer 1 state machine.
Events	An event that is an input to the layer 1 state machine.
All	All debug options

Examples To enable the ERRORS, INDICATIONS and EVENT debug options on all BRI interfaces, use the command sequence:

```
DISABLE BRI DEBUG=ALL
ENABLE BRI DEBUG=ERRORS
ENABLE BRI DEBUG=INDICATIONS
ENABLE BRI DEBUG=EVENTS
```

See Also DISABLE BRI DEBUG
SHOW BRI DEBUG

ENABLE BRI TEST

Syntax `ENABLE BRI=instance TEST=test-number`

where:

- *instance* is the number of the BRI interface.
- *test-number* is the number of the test to be enabled.

Description This command enables the specified test on the BRI interface. Only one test can be enabled on each invocation. Successive commands can be used to enable any combination of tests. The tests available depend on whether the BRI interface uses a MC145474 S/T interface controller (Table 4-21 on page 4-62), a PSB2186 S/T interface controller (Table 4-22 on page 4-63), a PEB2091 U interface controller (Table 4-23 on page 4-63) or a MC145572 U interface controller (Table 4-24 on page 4-64).

Table 4-21: ISDN Basic Rate Interface test modes for S/T interfaces using an MC145474 controller.

Test	Function
1	A loopback by the IMP of the data on the IDL bus towards the IMP.
2	A loopback by the IMP of the data on the IDL bus towards the interface.
3	A loopback by the transceiver of the B and D channel data on the IDL bus towards the IMP. Idles are transmitted on to the S/T loop.
4	A loopback by the transceiver of the B1 channel data on the IDL bus towards the IMP. Idles are transmitted on to the S/T loop in place of B1 data.
5	A loopback by the transceiver of the B2 channel data on the IDL bus towards the IMP. Idles are transmitted on to the S/T loop in place of B2 data.

Table 4-21: ISDN Basic Rate Interface test modes for S/T interfaces using an MC145474 controller. (Continued)

Test	Function
6	A loopback by the transceiver of the B1 channel data on the S/T loop towards the S/T loop. The data is also passed through to the IDL bus, but data received on the IDL bus for channel B1 is ignored.
7	A loopback by the transceiver of the B2 channel data on the S/T loop towards the S/T loop. The data is also passed through to the IDL bus. Data received on the IDL bus for channel B2 is ignored.
8	A loopback by the transceiver of the B1 channel data on the S/T loop towards the S/T loop. The data is not passed through to the IDL bus, idles are transmitted in its place. Data received on the IDL bus for channel B1 is ignored.
9	A loopback by the transceiver of the B2 channel data on the S/T loop towards the S/T loop. The data is not passed through to the IDL bus, idles are transmitted in its place. Data received on the IDL bus for channel B2 is ignored.
10	The transceiver will receive and demodulate its own transmitted data provided the transmit pair is connected to the receive pair at the interface connector. For this test to work correctly tests 12 and 15 should also be enabled.
11	A 96kHz test tone is transmitted on to the S/T loop.
12	The transceiver is forced into the highest INFO state, i.e. the transceiver transmits INFO 4 for a TE or INFO 3 for a NT.
13	The transceiver transmits without regard for the D channel contention procedures governing transmission. This test is applicable to a TE only.
14	The transceiver outputs E channel data on to the IDL bus in place of the D channel data received from the NT. This test is applicable to a TE only.
15	The transceiver will clock the IDL bus even if it is not able to derive a clock from the S/T loop. This test is applicable to a TE only.

Table 4-22: ISDN Basic Rate Interface test modes for S/T interfaces using a PSB2186 controller.

Test	Function
1	Single alternating pulses are sent at a 2kHz repetition rate.
2	Continuous alternating pulses are sent.
3	Data transmitted by the router is internally looped back to its receiver.
4	Data received at the interface is looped back out of the interface by the transceiver.

Table 4-23: ISDN Basic Rate Interface test modes for U interfaces using a PEB2091 controller.

Test	Function
1	Force a reset of the controller so that it enters quiet mode and does not transmit on the U loop.
2	Force the controller to transmit SN3 (standard framed, scrambled signal) on the U loop.
3	Enable an analogue loopback so that the router receives the data it transmits.
4	Enable the internal 2B + D test access port.

Table 4-23: ISDN Basic Rate Interface test modes for U interfaces using a PEB2091 controller. (Continued)

Test	Function
5	Enable a loopback of the B1 channel data on the U loop towards the U loop.
6	Enable a loopback of the B2 channel data on the U loop towards the U loop.
7	Enable a loopback of the B1, B2 and D channel data on the U loop towards the U loop.
8	Turn the activated LED on as soon as SN3 is transmitted to the LT, rather than when "act"=1 is received.

Table 4-24: ISDN Basic Rate Interface test modes for U interfaces using an MC145572 controller.

Test	Function
1	Force a reset of the controller so that it enters quiet mode and does not transmit on the U loop.
2	Force the controller to transmit SN3 (standard framed, scrambled signal) on the U loop.
3	Enable an analogue loopback so that the router receives the data it transmits.
4	Enable the internal 2B + D test access port.
5	Enable a loopback of the B1 channel data on the U loop towards the U loop and data transmitted by the router back to the router.
6	Enable a loopback of the B2 channel data on the U loop towards the U loop and data transmitted by the router back to the router.
7	Enable a loopback of the B1, B2 and D channel data on the U loop towards the U loop and data transmitted by the router back to the router.
8	Turn the activated LED on as soon as SN3 is transmitted to the LT, rather than when "act"=1 is received.
9	The interface will act as if it is an LT.



This command is required for testing only, and should not be used for normal operation of the BRI interface.

Examples To enable tests 8 and 9 on interface BRI0, use the commands:

```
DISABLE BRI=0 TEST
ENABLE BRI=0 TEST=8
ENABLE BRI=0 TEST=9
```

See Also DISABLE BRI CTEST
ENABLE BRI CTEST
DISABLE BRI TEST
SHOW BRI CTEST
SHOW BRI TEST

ENABLE ISDN CALL

Syntax `ENABLE ISDN CALL=name`

where:

- *name* is an ISDN call name, 1 to 15 characters in length. Valid characters are letters (a–z, A–Z), decimal digits (0–9) and underscore (“_”). It is case-insensitive.

Description This command enables an ISDN call definition. Existing active calls for this call are unaffected.

Examples To enable ISDN call “ROHO”, use the command:

```
ENABLE ISDN CALL=ROHO
```

See Also `ACTIVATE ISDN CALL`
 `ADD ISDN CALL`
 `DEACTIVATE ISDN CALL`
 `DELETE ISDN CALL`
 `DISABLE ISDN CALL`
 `SET ISDN CALL`
 `SHOW ISDN CALL`

ENABLE ISDN LOG

Syntax `ENABLE ISDN LOG`

Description This command enables the ISDN call logging facility. Call logging records details of events associated with ISDN calls.

When the ISDN logging facility is enabled, an entry is added to the log when a call is initiated. When the number of entries in the log reaches a limit, which is user-definable, the oldest entry in the log which represents a call that has completed is removed. If no entries represent completed calls, the log is allowed to grow beyond the limit.

When an ISDN call completes, either because the call was cleared in the setup phase, or because of normal call clearing, the log entry for the call is completed. At this time, a message similar to the message that appears in the ISDN log may be sent to an asynchronous port on the router. This port is user-definable.

The forwarding of ISDN log messages to the router’s logging facility is not affected by the status of the ISDN call logging facility.

Examples To enable ISDN call logging, use the command:

```
ENABLE ISDN LOG
```

See Also `DISABLE ISDN LOG`
 `DISABLE Q931 DEBUG`
 `ENABLE Q931 DEBUG`
 `SET ISDN LOG`
 `SHOW ISDN LOG`

ENABLE Q931 ASPID

Syntax `ENABLE Q931=interface ASPID=index[,index...]`

where:

- *interface* is a slotted interface name or number. Interface names are formed by concatenating an interface type and instance (e.g. BRI0 or PRI1). Interface numbers are the decimal index of the slotted interface (0, 1, 2...).
- *index* is a decimal number.

Description This command enables one or more auto-SPID values for a given Q.931 interface. When the auto-SPID process discovers more than one auto-SPID value, it puts them in a list. This list can be displayed with the SHOW Q931 SPID command on page 4-117. This command allows the user to select one or more of these values and enable them for use in the router.

The ASPID parameter selects which auto-SPID values are to be enabled. The numbers given in the ASPID parameter are the indices of the auto-SPID entries displayed with the SHOW Q931 SPID command on page 4-117.

Examples For the display shown in Figure 4-36 on page 4-119, to enable auto-SPID values whose indices are 1 and 2, enter the following:

```
ENABLE Q931=0 ASPID=1,2
```

See Also SET Q931
 SHOW Q931 SPID

ENABLE Q931 DEBUG

Syntax `ENABLE Q931=interface DEBUG={MDECODE|MRAW|SDLC|SINTERFACE|SSPID|SSPIDFILE|STATE|TRACE}`

where:

- *interface* is a slotted interface name or number. Interface names are formed by concatenating an interface type and instance (e.g. BRI0 or PRI1). Interface numbers are the decimal index of the slotted interface (0, 1, 2...).

Description This command enables the specified Q.931 debug option on the specified interface.

The MDECODE option displays each Q.931 message in a decoded format (Figure 4-6 on page 4-67, Table 4-25 on page 4-68). The message header is decoded, along with the type of each information element (IE) in the message. The octets of each IE are displayed in hexadecimal digit format, and for some IEs, a further decode is provided. If MDECODE debugging is enabled on an interface on which MRAW debugging is already enabled, then the MRAW display will be turned off, so that only the decoded messages will be displayed.

The MRAW debug option displays, for each Q.931 message sent or received on the specified interface, a display of the octets in the message, with no interpretation (Figure 4-7 on page 4-68, Table 4-26 on page 4-68). Each octet is displayed as two hexadecimal digits.

The SDLC option displays all DLC state machine events and state changes, as they occur, for the given interface (Figure 4-8 on page 4-69, Table 4-27 on page 4-69). The DLC state machine controls the activation and deactivation of DLCs on the Q.931 interface.

The SINTERFACE option displays all interface state machine events and state changes, as they occur, for the given interface. The interface state machine controls automatic switch detection as well as other procedures related to bringing an interface to a usable state.

The SSPID option displays all SPID state machine events and state changes, as they occur, for the given interface (Figure 4-9 on page 4-69, Table 4-28 on page 4-69). The SPID state machine controls the initialisation of the DLC via the SPID procedures, as well as auto-SPID detection.

The SSPIDFILE option displays all SPID file state machine events and state changes (Figure 4-10 on page 4-70, Table 4-29 on page 4-70). The SPID file state machine is concerned with controlling which SPID (generic, manual, auto-SPID) is actually used in initialising.

The STATE option displays all call state events and state changes (Figure 4-11 on page 4-70, Table 4-30 on page 4-71). The call state machine takes an ISDN call from initiation through to establishment to disconnection.

The TRACE option displays all subroutine calls within the Q.931 module. This option is not related to a particular interface, so while the interface must be entered as part of the command, subroutine tracing for all interfaces will be enabled.

Figure 4-6: Example output from the ENABLE Q931 DEBUG=MDECODE command for a call initiated by the router.

```

2454.5479 : LAPD OUT(I): Int: 0, DLC: 0
Protocol           08
Call reference     01 05
Message type       05 (SETUP)
IEs:
04 Bearer capability      02 88 90
18 Channel identification 01 83
70 Called party number    05 81 32 32 32 32
71 Called party subaddress 07 80 50 74 65 73 74 31

2454.7941 : LAPD IN(I): Int: 0, DLC: 0
Protocol           08
Call reference     01 85
Message type       07 (CONNECT)
IEs:
18 Channel identification 01 89

2455.2822 : LAPD OUT(I): Int: 0, DLC: 0
Protocol           08
Call reference     01 05
Message type       0f (CONNECT ACK)
IEs:

```

Table 4-25: Parameters displayed in the output of the ENABLE Q931 DEBUG=MDECODE command.

Parameter	Meaning
Timestamp	The time in seconds since the router restarted. This value rolls over at 9999 seconds.
LAPD IN, LAPD OUT	The direction of the message, with respect to the router.
(I), (U)	The type of message, numbered or unnumbered.
Int	The index of the ISDN interface over which the message was sent or received.
DLC	The index of the DLC over which this message was sent or received. If the message is an incoming UI frame, the DLC is "BROADCAST".
Protocol	The protocol ID field.
Call reference	The call reference field.
Message type	The type of the message.
IEs	The information elements in the message, one per line.

Figure 4-7: Example output from the ENABLE Q931 DEBUG=MRAW command for a call initiated by the router.

```

2454.5479 : LAPD OUT(I): Int: 0, DLC: 0
Data: 08 01 05 04 02 88 90 18 01 83 70 05 81 32 32 32 32 71 07 80 50 74 65
      73 74 31

2454.7941 : LAPD IN(I): Int: 0, DLC: 0
Data: 08 01 85 07 18 01 89

2455.2822 : LAPD OUT(I): Int: 0, DLC: 0
Data: 08 01 05 0f

```

Table 4-26: Parameters displayed in the output of the ENABLE Q931 DEBUG=MRAW command.

Parameter	Meaning
Timestamp	The time in seconds since the router restarted. This value rolls over at 9999 seconds.
LAPD IN, LAPD OUT	The direction of the message, with respect to the router.
(I), (U)	The type of message, numbered or unnumbered.
Int	The index of the ISDN interface over which the message was sent or received.
DLC	The index of the DLC over which this message was sent or received. If the message is an incoming UI frame, the DLC is "BROADCAST".
Data	The data in the message.

Figure 4-8: Example output from the ENABLE Q931 DEBUG=SDLC command.

```

3997.1924 Q931: DLC event - int=bri0, DLC=1, event=<Release Indication>,
state=<Established>
3997.1924 Q931: DLC state change - int=bri0, DLC=1, <Established> -> <Released>
4004.0777 Q931: DLC event - int=bri0, DLC=1, event=<Establish Indication>,
state=<Released>
4004.0777 Q931: DLC state change - int=bri0, DLC=1, <Released> -> <Established>
4007.0014 Q931: DLC event - int=bri0, DLC=1, event=<Establish>, state=<Established>

```

Table 4-27: Parameters displayed in the output of the ENABLE Q931 DEBUG=SDLC command.

Parameter	Meaning
Timestamp	The time in seconds since the router restarted. This value rolls over at 9999 seconds.
DLC event	A line of information about an event affecting the DLC state machine.
DLC state change	A line of information about a state change in the DLC state machine.
int	The name of the interface to which the event or state change applies.
DLC	The index of the DLC to which the event or state change applies.
event	The DLC event to which this message applies.
state	The DLC state when the event occurred.
<oldstate> -> <newstate>	The old and new states for a DLC state change.

Figure 4-9: Example output from the ENABLE Q931 DEBUG=SSPID command.

```

7110.5651 Q931: SPID event - int=bri0, DLC=1, event=<RESET>, state=<OP>
7110.5651 Q931: SPID state change - int=bri0, DLC=1, <OP> -> <NULL>
7110.5663 Q931: SPID event - int=bri0, DLC=1, event=<INIT>, state=<NULL>
7110.5663 Q931: SPID state change - int=bri0, DLC=1, <NULL> -> <IWAIT1>
7110.6428 Q931: SPID event - int=bri0, DLC=1, event=<INFO>, state=<IWAIT1>
7110.6428 Q931: SPID state change - int=bri0, DLC=1, <IWAIT1> -> <OP>

```

Table 4-28: Parameters displayed in the output of the ENABLE Q931 DEBUG=SSPID command.

Parameter	Meaning
Timestamp	The time in seconds since the router restarted. This value rolls over at 9999 seconds.
SPID event	A line of information about an event that affects the SPID state machine.
SPID state change	A line of information about a state change in the SPID state machine.
int	The name of the interface to which the event or state change applies.
DLC	The index of the DLC to which the event or state change applies.

Table 4-28: Parameters displayed in the output of the ENABLE Q931 DEBUG=SSPID command. (Continued)

Parameter	Meaning
event	The SPID event to which this message applies.
state	The SPID state when the event occurred.
<oldstate> -> <newstate>	The old and new states for a SPID state change.

Figure 4-10: Example output from the ENABLE Q931 DEBUG=SSPIDFILE command.

```

7380.2693 Q931: SPID file event: int=bri0, state=1, event=SetSPID
7380.2693 Q931: SPID file state change: int=bri0, state=1 -> 1
7380.4200 Q931: SPID file event: int=bri0, state=1, event=ConfSPIDPass
7380.4200 Q931: SPID file state change: int=bri0, state=1 -> 11

```

Table 4-29: Parameters displayed in the output of the ENABLE Q931 DEBUG=SSPIDFILE command.

Parameter	Meaning
Timestamp	The time in seconds since the router restarted. This value rolls over at 9999 seconds.
SPID file event	A line of information about an event that affects the SPID file state machine.
SPID file state change	A line of information about a state change in the SPID file state machine.
int	The name of the interface to which the event or state change applies.
state	The SPID file state when the event occurred.
event	The SPID file event to which this message applies.
state=n -> m	The old and new states for a SPID file state change.

Figure 4-11: Example output from the ENABLE Q931 DEBUG=STATE command.

```

7547.0903 Q931: Call event - int=bri0, call=1, state 0, event 77
7547.0903 Q931: Call state change - int=bri0, call=1, 0 -> 1
7547.1561 Q931: Call event - int=bri0, call=1, state 1, event 43
7547.1561 Q931: Call state change - int=bri0, call=1, 1 -> 2
7548.2902 Q931: Call event - int=bri0, call=1, state 2, event 71
7548.2902 Q931: Call state change - int=bri0, call=1, 2 -> 11
7548.3480 Q931: Call event - int=bri0, call=1, state 11, event 38
7548.3480 Q931: Call state change - int=bri0, call=1, 11 -> 0

```


Table 4-30: Parameters displayed in the output of the ENABLE Q931 DEBUG=STATE command.

Parameter	Meaning
Timestamp	The time in seconds since the router restarted. This value rolls over at 9999 seconds.
Call event	A line of information about an event that affects the call state machine.
Call state change	A line of information about a state change in the call state machine.
int	The name of the interface to which the event or state change applies.
call	The index of the call to which the event or state change applies.
state	The call state when the event occurred.
event	The call event to which this message applies.
n -> m	The old and new states for a call state change.

Examples To enable the display of debugging information for auto-SPID detection and SPID initialisation on Q.931 interface 1, use the commands:

```
ENABLE Q931=1 DEBUG=SSPID
ENABLE Q931=1 DEBUG=SSPIDFILE
```

See Also DISABLE ISDN LOG
DISABLE Q931 DEBUG
ENABLE ISDN LOG
SET ISDN LOG
SHOW ISDN LOG

ENABLE RAPI

Syntax ENABLE RAPI

Description This command enables Remote CAPI (RAPI). The router will establish sessions to RAPI clients and interact with RAPI clients using Device Control Protocol (DCP) messages. RAPI is disabled by default and must be enabled to support RAPI clients.

Examples To enable remote CAPI, use the command:

```
ENABLE RAPI
```

See Also DISABLE RAPI

RESET BRI

Syntax RESET BRI=*instance*

where:

- *instance* is the number of the BRI interface.

Description This command resets the BRI interface. The hardware is reset but configuration information is retained. The S/T loop activation procedure is re-initiated.



At present there is no known circumstance where use of this command is required and it should be used only under advice from the manufacturer.

Examples To reset BRI interface 0, use the command:

```
RESET BRI=0
```

See Also RESET BRI COUNTERS
SHOW BRI STATE

RESET BRI COUNTERS

Syntax RESET BRI [= *instance*] COUNTERS [= { INTERFACE | BRI }]

where:

- *instance* is the number of the BRI interface.

Description This command resets the BRI interface counters displayed by the SHOW BRI COUNTERS command on page 4-88. The counters are copied, and the values subtracted from the counter values whenever the counters are displayed by the SHOW BRI COUNTERS command. This gives the illusion of resetting the counters without affecting the MIB variables. If the BRI interface is not specified, the counters for all BRI interfaces are reset. If the counter category is not specified, all categories are reset.

Examples To reset the interface counters for BRI interface 0, use the command:

```
RESET BRI=0 COUNTERS=INTERFACE
```

See Also RESET BRI
SHOW BRI COUNTERS

RESET Q931

Syntax RESET Q931=*interface* [CALL={*call-index*|ALL}]

where:

- *interface* is a slotted interface name or number. Interface names are formed by concatenating an interface type and instance (e.g. BRI0 or PRI1). Interface numbers are the decimal index of the slotted interface (0, 1, 2...).
- *call-index* is the Q.931 index of an active ISDN call.

Description This command resets the specified Q.931 interface or call(s). The reset occurs using special Q.931 reset procedures, rather than the call control procedures used to deactivate ISDN calls.

The CALL parameter specifies the index of a Q.931 call. This index can be read from the Index field of the output of the SHOW Q931 command on page 4-114. The call index as obtained from the SHOW ISDN CALL command on page 4-103 should **not** be used in this command.

If a Q.931 call index is specified, a RESTART message specifying the channel used by the call is sent to the network, to reset that call only. If the call index ALL is specified, the RESTART message will specify all channels used by all calls. If the Q.931 call index is not specified, a RESTART message is sent to the network that indicates that the whole interface should be reset.

Examples To reset Q.931 interface 0, use the command:

```
RESET Q931=0
```

See Also SET Q931
SHOW Q931

SET BRI

Syntax SET BRI=*instance* [ACTIVATION={NORMAL|ALWAYS}]
[ISDNSLOTS=*slot-list*] [MODE={ISDN|TDM|MIXED}]
[TDMLOTS=*slot-list*]

where:

- *instance* is the number of the BRI interface.
- *slot-list* is a character string defining a list of slots. It may include the numbers 1 and 2 corresponding to the BRI slots B1 and B2. If both are specified they should be separated by a comma or a dash.

Description This command sets the values of the user-configurable BRI operational parameters.

The ACTIVATION parameter controls the operation of the layer 1 state machine. If NORMAL is specified, the state machine provides the standard mode of ISDN operation. If ALWAYS is specified the interface is assumed to be connected to a link that is expected to be active at all times. When the link is not

active the router will not attempt to activate the link by sending INFO 1. The default is NORMAL.

The ISDNSLOTS parameter specifies which of the slots are available for use by ISDN calls, and is only allowed when the MODE parameter is set to ISDN or MIXED. It is not permitted when MODE is set to TDM. The ISDNSLOTS parameter can be used to disable some slots, providing support for non-standard ISDN services, such as German Monopol. Slot numbers 1 and 2 correspond to the B1 and B2 slots, respectively. The default is for all slots to be available for ISDN calls.

The MODE parameter specifies the operational mode of the BRI interface. If ISDN is specified, a corresponding LAPD and Q.931 instance will exist and it will not be possible to create TDM groups on the port. The port is managed by ISDN call control, and higher layer modules access the port via an ISDN call. If TDM is specified, there will be no LAPD or Q.931 instances for the port. In this case, higher layer modules must access the port via a PPP interface configured directly to a TDM group that has been created to use some of the slots of the port. ISDN call control has no effect on the port when it is in TDM mode. If MIXED is specified then there will be LAPD and Q.931 instance for the port and the port may be used for ISDN calls. However, some of the port slots are available for TDM groups. The slots (B1 and B2) are apportioned between ISDN calls and TDM groups using the ISDNSLOTS and TDMSLOTS parameters. The default is ISDN.



The MODE parameter of the SET BRI command affects the way the router behaves when connected to a network to the extent that, if configured inappropriately for the network to which it is connected, it may not conform to the national standards applying to that network. Therefore care must be taken when using this command. Please seek the advice of your distributor or ISDN service provider when changing the mode of operation from the default, which is the correct mode for connecting to a standard ISDN network.



Semipermanent connections are not available in the USA and the router will not permit the MODE of a BRI U interface to be set TDM or MIXED or the ACTIVATION mode set to ALWAYS.

The TDMSLOTS parameter specifies which of the slots are available for use by TDM groups, and is only allowed when the MODE parameter is set to TDM or MIXED. It is not permitted when MODE is set to ISDN. The TDMSLOTS parameter can be used to restrict the use of slots by TDM groups when the interface is used for semipermanent connections. When the MODE parameter is set to MIXED, the default is for no slots to be available for TDM groups. When the MODE parameter is set to TDM, the default is for all slots to be available for TDM groups.

Examples To configure BRI interface 0 to use the B1 channel for ISDN calls and the B2 channel for a semipermanent connection, use the command:

```
SET BRI=0 MODE=MIXED ISDNSLOTS=1 TDMSLOTS=2
```

See Also RESET BRI
SHOW BRI STATE

SET ISDN CALL

Syntax SET ISDN CALL=*name* [NUMBER=*number*] [PRECEDENCE={IN|OUT}] [ALTNUMBER=*number*] [BUMPDELAY=0..100] [CALLBACK={ON|OFF|YES|NO|TRUE|FALSE}] [CALLINGNUMBER=*number*] [CALLINGSUBADDRESS=*calling-subaddress*] [CBDELAY=0..100] [CHECKCLI={OFF|PRESENT|REQUIRED}] [CHECKSUB={OFF|LOCAL|REMOTE}] [CHECKUSER={OFF|LOCAL|REMOTE}] [CLILIST=0..99] [DIRECTION={IN|OUT|BOTH}] [DOV={ON|OFF|YES|NO|TRUE|FALSE}] [HOLDUP=0..7200] [INANY={ON|OFF|YES|NO|TRUE|FALSE}] [INTPREF={NONE|*interface*}] [INTREQ={NONE|*interface*}] [KEEPUP={ON|OFF|YES|NO|TRUE|FALSE}] [LOGIN={ALL|NONE|CHAP|PAP-RADIUS|PAP-TACACS|USER}] [OUTCLI={OFF|CALLING|INTERFACE|NONUMBER}] [OUTSUB={OFF|LOCAL|REMOTE}] [OUTUSER={OFF|LOCAL|REMOTE}] [PASSWORD={NONE|CLI|CALLED|SUB|NAME|USER}] [PPPTEMPLATE=*template*] [PRIORITY=0..99] [RATE={56K|64K}] [REMOTECALL=*name*|*remote-number*] [RN1=0..10] [RN2=0..5] [RT1=5..120] [RT2=300..1200] [SEARCHCLI={ON|OFF|YES|NO|TRUE|FALSE|CALLED|0..99}] [SEARCHSUB={OFF|LOCAL|REMOTE}] [SEARCHUSER={OFF|LOCAL|REMOTE}] [SUBADDRESS=*number*] [USER={ATTACH|PPP}] [USERNAME={NONE|CLI|CALLED|SUB|NAME|USER}]

where:

- *name* is an ISDN call name, 1 to 15 characters in length. Valid characters are letters (a–z, A–Z), decimal digits (0–9), hyphen (“-”) and underscore (“_”). It is case-insensitive.
- *number* is an ISDN phone number, 1 to 31 characters in length. Valid characters are decimal digits (0–9).
- *calling-subaddress* is a character string, 1 to 31 characters in length. Valid characters are letters (a–z, A–Z), decimal digits (0–9) and underscore (“_”). It is case-insensitive.
- *interface* is a slotted interface name or number. Interface names are formed by concatenating an interface type and instance (e.g. BRI0 or PRI1). Interface numbers are the decimal index of the slotted interface (0, 1, 2...).
- *template* is a number in the range 0 to 31.
- *remote-number* is a number, 1 to 15 characters in length. Valid characters are decimal digits (0–9).

Description This command modifies an ISDN call definition. The call definition must already exist. Any active calls associated with this call definition are unaffected, but new calls made using this definition will use the new parameters.

The CALL parameter uniquely identifies this call in the router. All commands that affect ISDN call definitions must specify the call with this parameter. ISDN call names are case-insensitive. The case of the ISDN call name as entered will be saved, so case can be used to provide readable names. However, any form of the name can be used in subsequent commands, and no two calls may have the same name when case is ignored.

The NUMBER parameter specifies the number called when this call is activated. This is the number that Q.931 uses in the SETUP message passed to the network, so it must include all access and area codes required by the

network and be formatted in the way required by the network. Spaces or other characters may not be entered in between the digits of the number.

The PRECEDENCE parameter specifies the direction of precedence for the call in the event of call collision. Call collision occurs when a call is activated at the same time as an incoming call selects the same call. If precedence is IN, the incoming call has precedence and the outgoing call is cleared. If precedence is OUT, the outgoing call has precedence and the incoming call is cleared.

The ALTNUMBER parameter specifies an alternate ISDN number for this call to ring if all retries and retry groups for the main number have failed. The ISDN call retry parameters (RN1, RN2, RT1 and RT2) apply only to the main ISDN number. The alternate number is tried only once. The KEEPU parameter, if set, forces ISDN call control to cycle repeatedly through the main number, all retries and retry groups for the main number, and then the alternate number, until a call succeeds.

The BUMPDELAY parameter specifies the time, in tenths of a second, the router will wait after bumping another call before initiating this call. Call bumping involves clearing a call and using the resulting free B channel for a new call. A delay is programmable with the BUMPDELAY parameter in order to give the network time to clear the bumped call's B channel. The default is 5, that is, 0.5s.

The CALLBACK parameter specifies whether this call, upon being selected by an incoming call, should clear the incoming call and call back or not. The values ON, TRUE and YES are equivalent and mean that the call back will occur. The values OFF, FALSE and NO are equivalent, and mean that the call back will not occur. The default value is NO.

The CALLINGNUMBER parameter may be used in connecting this call to a remote call. Certain options for formatting the outgoing SETUP message allow the calling number to be specified.

The CALLINGSUBADDRESS parameter specifies a calling subaddress to be placed in the outgoing SETUP message. This value will be placed in the outgoing SETUP message only if the OUTCLI parameter is set to CALLING.

The CBDELAY parameter specifies the time, in tenths of a second, the router will wait after clearing a call before initiating a callback for the call. Call back involves clearing a call and using the resulting free B channel for the new call. A delay is programmable with the CBDELAY parameter in order to give the network time to clear the incoming call's B channel. The default is 41, that is, 4.1s.

The CHECKCLI parameter specifies how this call, if selected, is checked against the CLI IE in the incoming SETUP message. The check, if carried out, consists of verifying that the CLI number appears in the CLI list for this call. The default value of OFF means that no check is carried out. The value PRESENT means that the check is carried out only if the CLI IE is present, and contains calling number digits. The check passes if the CLI IE is not present, or does not contain calling number digits, or is present and contains a matching CLI number. The value REQUIRED means that CLI MUST be present, and must contain calling number digits. The check fails if the CLI IE is not present, or does not contain calling number digits, or does not contain a matching CLI number.

The CHECKSUB parameter specifies whether this call, when selected, should have the called party subaddress IE of the incoming SETUP message checked.

The IE may be checked against the call name (parameter set to LOCAL) or the remote call name (parameter set to REMOTE). The default value is OFF, which means that no check is carried out.

The CHECKUSER parameter specifies whether this call, when selected, should have the user-user data IE of the incoming SETUP message checked. The IE may be checked against the call name (parameter set to LOCAL) or the remote call name (parameter set to REMOTE). The default value is OFF, which means that no check is carried out.

The CLILIST parameter specifies the CLI list against which this call is checked if the check CLI parameter is either PRESENT or REQUIRED. The default value is a special value that means that the list is undefined.

The DIRECTION parameter specifies the directions for which the call is enabled. Calls may be enabled both for sending and receiving calls, or for either direction. The default value is BOTH.

The DOV parameter specifies whether or not the outgoing call setup message for this call has data bearer capability or voice bearer capability. If DOV is set to ON, voice bearer capability is specified and the ISDN service will treat the call as a voice call. If DOV is set to OFF, data bearer capability is specified and the ISDN service will treat the call as a data call. The values ON and TRUE are equivalent to YES. The values OFF and FALSE are equivalent to NO. The default is NO. The DOV parameter is used in conjunction with the DOVNUMBER parameter on the SET Q931 command on page 4-84 to configure data over voice (DOV).

The HOLDUP parameter specifies the minimum time, in seconds, that this call should be held up after activation. If the user of the ISDN call requests a deactivation, and the holdup time has not expired, the deactivation will be ignored until the holdup time has expired. The default for this parameter is 0 seconds.

The INANY parameter specifies whether this call may be selected to match any incoming call. The search for calls with INANY set YES follows all other searches. Only one call should have INANY set to YES, since otherwise a predictable response to incoming calls cannot be guaranteed. The default value for this parameter is NO.

The INTREQ parameter specifies which ISDN interface MUST be used for this call, when the call is activated as an outgoing call. If no channel is available on the required interface, the call will fail. The default for this parameter is NONE, which means no required interface.

The INTPREF parameter specifies which ISDN interface should preferentially be used for this call, if the required interface is not specified. When activating this call, the preferred interface is checked first for a free channel. If no free channel is found, other interfaces may be checked. The default for this parameter is NONE, which means no preferred interface.

The KEEPU parameter determines whether the call should be kept up at all costs or not. The KEEPU parameter for a call is inspected when all retries for the main number have failed and the alternate number (if defined) has also failed, and when the call is cleared for any reason other than explicit clearing by the user module or by manager command. If the KEEPU parameter has the value YES, the call will be reactivated in these circumstances. The values ON and TRUE are equivalent to YES. The values OFF, FALSE and NO are equivalent for turning off the KEEPU parameter. The default value is NO.

The LOGIN parameter specifies which login procedure this call must use when it is activated. If CHAP is specified, the call will be accepted but will create a PPP interface which will authenticate using CHAP. If PAP-RADIUS is specified, the call will be accepted but will create a PPP interface which will authenticate using PAP, and using RADIUS as the means of authenticating the PAP exchange. If PAP-TACACS is specified, the call will be accepted but will create a PPP interface which will authenticate using PAP, and using TACACS as the means of authenticating the PAP exchange. If USER is specified, the User Authentication Database in the router is checked. The default is NONE, which means that no login procedure is required.

The values CHAP, PAP-TACACS and PAP-RADIUS are only used when the ISDN call creates a dynamic PPP interface. Since these parameters can also be set by defining a PPP template with the appropriate authentication parameters, use of these values is for backward compatibility only. The value specified in the LOGIN parameter will override the authentication settings in the PPP template.

The OUTCLI parameter specifies the format of the calling party number IE and calling subaddress IE (also known as CLI) in the outgoing SETUP message created when this call is activated. If OFF is specified, the CLI is not included in the SETUP message. If CALLING is specified, the calling number and calling subaddress values from the ISDN call definition are placed in the SETUP message. If the CALLINGSUBADDRESS parameter is not defined, the calling subaddress IE will not be included in the SETUP message. If INTERFACE is specified, the number and subaddress values from the Q.931 interface (set with the SET Q931 command on page 4-84) are placed in the SETUP message. If the Q.931 interface does not have a subaddress set, the calling subaddress IE will not be included in the SETUP message. If NONUMBER is specified, an empty calling number IE and the calling subaddress from the Q.931 interface (if set) are included in the SETUP message. The ISDN itself can fill in the calling number IE in the SETUP message before sending the message to the remote end. The default is OFF.

The OUTSUB parameter specifies the format of the called party subaddress IE in the outgoing SETUP message created when this call is activated. The default value for this parameter is OFF, which means that the called party subaddress IE is not included in the SETUP. The call name or remote call name may be specified.

The OUTUSER parameter specifies the format of the user-user data IE in the outgoing SETUP message created when this call is activated. The default value for this parameter is OFF, which means that the user-user data IE is not included in the SETUP. The call name or remote call name may be specified.

The PASSWORD parameter specifies the source of the password for login procedures. The default value of NONE means that no password is specified. The values CLI, CALLEDSUB and USER mean that the password is drawn from, respectively, the CLI, called party subaddress and user-user data IE in the incoming SETUP message. The value of NAME means that the call name is used as the password.

The PPPTEMPLATE parameter specifies the PPP template to use when creating a dynamic PPP interface for this call. The specified template must exist. This parameter is only valid if encapsulation is set to AUTO, OKPPP or PPP. See "Templates" on page 3-9 of *Chapter 3, Point-to-Point Protocol (PPP)* for more information about creating PPP templates.

The PRIORITY parameter specifies the priority of this call for use by the call bumping facility. The value of this parameter is a number in the range 0 to 99. The default is 50. Table 4-15 on page 4-25 details how the different priority values affect the bumping of data calls.

The RATE parameter specifies the rate of data transmitted and received on the B channel for this call. The rate can be either 64 kbps (the default value) which is the full bandwidth of the B channel, or 56 kbps, which is specified by ITU-T standard V.110 (rate adaption). The data rate specified by this parameter will be used when this call is used as an outgoing call. When the call is selected as an incoming call, the data rate is determined by the bearer capability in the SETUP message or the rate set for the entire Q.931 interface, as specified by the SET Q931 command on page 4-84.

The REMOTECALL parameter is used to connect this call to a remote call. Some options for formatting the outgoing SETUP message and searching for calls allow the remote call to be specified. This parameter has the same syntax as the CALL parameter.

The RN1 parameter specifies how many times this call will be retried in a single retry group. The default value of 0 means that the call will not be retried in a retry group.

The RN2 parameter specifies how many retry groups this call will have, after the first group. The default value of 0 means that the first group only will be tried.

The RT1 parameter specifies the time in seconds between retries in the same retry group. The default is 30 seconds.

The RT2 parameter specifies the time in seconds between retry groups. The default is 600 seconds.

The SEARCHCLI parameter specifies whether this call may be included in a search based on the CLI IE in the incoming SETUP message. If ON is specified, the value of the CLI IE in the incoming SETUP message is compared with the called number (NUMBER) parameter of this call definition. The options TRUE, YES and CALLED are synonyms for ON. If OFF is specified, there is no search based on the CLI IE. The options FALSE and NO are synonyms for OFF. If a number is specified it identifies an existing CLI list, and the value of the CLI IE is compared with all numbers in the specified CLI list. The default value is OFF.

The SEARCHSUB parameter specifies whether this call may be included in a search based on the called party subaddress IE in the incoming SETUP message. In such a search, the called party subaddress IE may be compared with the call name (parameter set to LOCAL) or the remote call name (parameter set to REMOTE). The default value is OFF.

The SEARCHUSER parameter specifies whether this call may be included in a search based on the user-user data IE in the incoming SETUP message. In such a search, the user-user data IE may be compared with the call name (parameter set to LOCAL) or the remote call name (parameter set to REMOTE). The default value is OFF.

The SUBADDRESS parameter allows the specification of an entirely numeric subaddress to be placed in the outgoing SETUP message when this call is activated. The subaddress as specified by the OUTSUB parameter has the limitation that it can only be the remote or local call name, which means that entirely numeric subaddresses cannot be specified with this parameter alone.

However, in some cases, a numeric subaddress is required to satisfy network requirements when calling a router which shares an S/T bus with other ISDN devices. The default value for this parameter is a null (empty) string. If this parameter has a value, it overrides the OUTSUB parameter when setting the called subaddress IE in the outgoing SETUP message.

The USER parameter specifies how users of ISDN calls use this call. The value ATTACH, the default, means that users must attach to this call before it can be used. The value PPP means that this call is able to create dynamic PPP interfaces when activated. The PPP value is most likely to be used for incoming ISDN calls, which use the user data base to set parameters for the PPP and IP interfaces dynamically created.

The USERNAME parameter specifies the source of the user name for login procedures. The default value of NONE means that no user name is specified. The values CLI, CALLED SUB and USER mean that the user name is drawn from, respectively, the CLI, called party subaddress and user-user data IE in the incoming SETUP message. The value of NAME means that the call name is used as the user name.

Examples To enable the callback option and set the delay between clearing the call and calling back to 2 seconds for ISDN call "Region-1", use the command:

```
SET ISDN CALL="Region-1" CALLBACK=ON CBDELAY=20
```

See Also ACTIVATE ISDN CALL
ADD ISDN CALL
DEACTIVATE ISDN CALL
DELETE ISDN CALL
DISABLE ISDN CALL
ENABLE ISDN CALL
SHOW ISDN CALL

SET ISDN DOMAINNAME

Syntax SET ISDN DOMAINNAME=*domain-name*

where:

■ *domain-name* is a domain name.

Description This command modifies the domain name to be prepended to a login name for a DNS lookup to determine the IP address to be used for an ISDN call. Only one domain name may be defined.

Examples To change the domain name "acc.newco.co.nz" to "sales.southern.com" for use with DNS lookups, use the command:

```
SET ISDN DOMAINNAME=sales.southern.com
```

See Also ADD ISDN DOMAINNAME
DELETE ISDN DOMAINNAME
SHOW ISDN DOMAINNAME

SET ISDN LOG

Syntax `SET ISDN LOG [PORT={0..23|NONE}] [LENGTH=0..100]`

Description This command sets parameters for the ISDN call logging facility. Call logging records details of events associated with ISDN calls.

The PORT parameter specifies the asynchronous port on the router to which ISDN log messages are sent. The messages are sent when the log entry reaches a completed state, which means that the call has been cleared, either in the setup phase, or as a result of normal call clearing. Setting the login port to NONE, the default value, disables the sending of messages to any asynchronous port on the router.

The LENGTH parameter specifies the maximum length, in number of entries, of the ISDN call log. The default is 25.

Examples To set the ISDN call log to its maximum length but not output any messages to a terminal, use the command:

```
SET ISDN LOG PORT=NONE LENGTH=100
```

To view the ISDN call log, use the command:

```
SHOW ISDN LOG
```

See Also `DISABLE ISDN LOG`
 `DISABLE Q931 DEBUG`
 `ENABLE ISDN LOG`
 `ENABLE Q931 DEBUG`
 `SHOW ISDN LOG`

SET LAPD

Syntax SET LAPD=*interface* DEBUG={OFF|STATE|PACKET}

SET LAPD=*interface* MODE={AUTOMATIC|NONAUTOMATIC}

SET LAPD=*interface* [NASMODE={NORMAL|MASTER|SLAVE}]
[NASMASTER=*interface*]

SET LAPD=*interface* SAP=*sap* K=*value*

SET LAPD=*interface* SAP=*sap* {N200|N201|N202}=*time*...

SET LAPD=*interface* SAP=*sap* {T200|T201|T202|T203}=*time*...

The following variants may be used for conformance testing only:

```
SET LAPD=interface {ATTACH=sap|CONNECT=sap|DATA=sap
CES=ces|ESTABLISH=sap CES=ces|MDATA=sap CES=ces|
MUNIT=sap CES=ces|RELEASE=sap CES=ces|UNIT=sap CES=ces}
```

where:

- *interface* is a slotted interface number (0, 1, 2...).
- *sap* is a SAP identifier.
- *ces* identifies a DLC within the SAP.
- *time* is a time value in tenths of a second.

Description This command configures LAPD on an ISDN interface.

The DEBUG parameter controls the display of debug messages. If OFF is specified, debug messages are not displayed. If STATE is specified, a message is displayed each time the LAPD interface experiences a state transition. If PACKET is specified, every LAPD packet received on the interface is decoded and displayed.



The DEBUG option is only required for testing, and should not be used in normal operation. Due to the volume of output, it may be difficult to turn off and will also result in reduced router performance.

The MODE specifies the TEI assignment mode. If AUTOMATIC or NONAUTOMATIC is specified, automatic TEI assignment is enabled or disabled, respectively. The default is AUTOMATIC.

The NASMODE parameter specifies the non-associated signalling mode for this interface. The value NORMAL specifies that D channel signalling for the calls on this interface will take place on this interface's D channel, and this interface's D channel will not provide the signalling for any other ISDN interface. The value MASTER specifies that D channel signalling for the calls on this interface will take place on this interface's D channel, and that this interface's D channel may provide signalling for other ISDN interfaces. The value SLAVE specifies that D channel signalling for the calls on this interface will take place on the D channel of another ISDN interface. If the value SLAVE is specified, the NASMASTER parameter must be present on the same command line. The default value for this parameter is NORMAL.

The NASMASTER parameter specifies the ISDN interface whose D channel will provide signalling for calls on this interface, if the NASMODE for this interface is SLAVE. The NASMODE of the interface specified with the NASMASTER parameter must be MASTER. There is no default value for this parameter, since the default NASMODE is NORMAL.

The SAP parameter specifies the SAP identifier of the SAP to be modified..

The K parameter specifies the number of outstanding I frames allowed.

The N200, N201, N202, T200, T201, T202 and T203 parameters specify the value (in tenths of a second) of the respective timer.



The k, N2xx and T2xx parameters must conform to the LAPD standard.

The ATTACH, CONNECT, ESTABLISH, RELEASE, DATA, MDATA, UNIT and MUNIT options are only required for conformance testing and should not be used in normal operation.

The ATTACH parameter adds a SAP to a LAPD interface. The CONNECT parameter adds a DLC to a SAP. The ESTABLISH parameter attempts to establish a DLC using the value *ces* returned from a previous ATTACH command. The RELEASE parameter de-establishes a DLC. The DATA parameter sends an I frame. The MDATA parameter sends 16 I frames. The UNIT parameter sends a UI frame. The MUNIT parameter sends 16 UI frames.



Only one of these options should be entered per command.

Examples

To enable debugging of all LAPD packets on LAPD interface 0, use the command:

```
SET LAPD=0 DEBUG=PACKET
```

See Also

SHOW LAPD

SET Q931

Syntax SET Q931=*interface* [DOVNUMBER=*number*] [INTID=[*hex-string*]]
 [NONUM={ACCEPT|REJECT}] [NOSUB={ACCEPT|REJECT}]
 [NUM1=*number*] [NUM2=*number*] [PROFILE={5ESS|AUS|CHINA|
 DMS-100|ETSI|JAPAN|KOREA|NI1|NZ}] [RATE={56K|64K}]
 [SPID1=*spid*] [SPID2=*spid*] [SUB1=*subaddress*]
 [SUB2=*subaddress*] [*timer*={OFF|*time*}]

where:

- *interface* is a slotted interface name or number. Interface names are formed by concatenating an interface type and instance (e.g. BRI0 or PRI1). Interface numbers are the decimal index of the slotted interface (0, 1, 2...).
- *hex-string* is a string, 1 to 16 characters in length. Valid characters are decimal digits (0–9) and the letters a–f or A–F.
- *number* is an ISDN phone number, 1 to 39 characters in length. Valid characters are decimal digits (0–9).
- *spid* is an ISDN Service Provider Identifier, 1 to 31 characters in length. Any character is valid, although decimal digits (0–9) will almost invariably be the only characters used.
- *subaddress* is an ISDN subaddress, 1 to 39 characters in length. Valid characters are decimal digits (0–9), uppercase letters (A–Z) and lowercase letters (a–z). It is case-insensitive.
- *timer* is the name of a Q.931 timer, and must be one of T301, T302, T303, T304, T305, T308, T309, T310, T313, T314, T316, T317, T318, T319, T321 or T322.
- *time* is the timeout value for the timer.

Description This command configures the Q.931 module on an ISDN interface.

The DOVNUMBER parameter specifies an ISDN number for the interface. If a call is received on this interface with a voice bearer capability and a called number matching the value specified for DOVNUMBER, the call is treated as a data call, not a voice call.

The INTID parameter specifies an interface identifier for use in non-associated signalling. Non-associated signalling is configured with the SET LAPD command on page 4-82; this parameter just sets the identifier the network and router use to distinguish between the interfaces sharing a common D channel for signalling purposes. The parameter is a sequence of hexadecimal digits which give the hexadecimal representation of the interface identifier. Since the interface identifier is given by the ISDN network provider as part of the subscription process to the non-associated signalling feature, the provider must have made clear the exact format of the interface identifier. Entry as a hexadecimal string allows any sequence of bits to be specified as an interface identifier, but some conversion may be required. For example, if the interface identifier is given as the sequence of characters, "I1", it will have to be converted to hexadecimal and entered as INTID=4931.

The NONUM parameter specifies the behaviour of the router towards an incoming call that does not contain a called number. The router can be set up to either reject these calls, or accept them, given that other conditions allow the call to be accepted. In most ISDNs there will always be a called number present in an incoming call. The default is ACCEPT.

The NOSUB parameter specifies the behaviour of the router towards an incoming call that does not contain a called subaddress. The router can be set up to either reject these calls, or accept them, given that other conditions allow the call to be accepted. Most ISDNs will only present a called subaddress in an incoming call if the remote user sent a called subaddress, so setting this parameter to REJECT could have undesirable results. The default is ACCEPT.

The NUM1 and NUM2 parameters assign the router's own ISDN phone numbers. These parameters are only required when the router is attached to a BRI S/T bus with other TEs, or if SPIDS have been defined. If the router is the only TE on the bus, all incoming calls will be for the router. If more than one TE exists on the bus, the incoming setup message is sent to all of them, and the called number in the setup message must be matched with the TE's number before it may reply to the call. The number entered should be the number as supplied by the carrier, without STD access codes or area codes. The incoming number and the router's number will be compared from the right-hand end and only as far as the shortest of the two.

The PROFILE parameter determines which network is running on the interface (Table 4-31 on page 4-85). The profile is set automatically whenever the router territory is changed by the SET SYSTEM TERRITORY command on page 1-56 of *Chapter 1, Operation*. The default territory is 'Europe' which sets the profile to ETSI.

Table 4-31: Q.931 Profiles.

Profile Name	Access Mode	Country
5ESS	Basic Rate	5ESS custom, USA
AUS	Basic or Primary Rate	Australian Telecom
CHINA	Basic or Primary Rate	China Telecom
DMS-100	Basic rate	DMS-100 custom, USA.
ETSI	Basic or Primary Rate	European Union (EU) and European Free Trade Association (EFTA) countries—ETSI specification
JAPAN	Basic or Primary Rate	Japan
KOREA	Basic or Primary Rate	Korea
NI1	Basic Rate	National ISDN, USA
NZ	Basic or Primary Rate	New Zealand Telecom



If you are not sure about which profile to use, contact your distributor or ISDN service provider.



Failure to select the correct profile will invalidate the approval of this product with respect to the applicable national standards for the country in which the product is used.

The RATE parameter specifies the rate of data transmitted and received on the B channel for all calls in this interface. The rate can be either 64 kbps (the default value) which is the full bandwidth of the B channel, or 56 kbps, which is specified by ITU-T standard V.110 (rate adaption). All calls made and received on this interface will use the rate specified by this parameter.

The SPID1 and SPID2 parameters specify Service Provider Identifiers for the router. These will not be needed in most cases, but where required, the ISDN service provider will supply the values.

The SUB1 and SUB2 parameters specify the router's ISDN subaddresses. These parameters are only required when the router is attached to a BRI S/T bus with other TEs. If the router is the only TE on the bus, all incoming calls will be for the router. If more than one TE exists on the bus, the incoming setup message is sent to all of them, and the subaddress in the SETUP message must match the TE's subaddress before it may reply to the call. If neither subaddress of a Q931 interface is set, the subaddress in the SETUP message will be passed to call control for processing as part of an ISDN call. The subaddresses as set by this command can match the subaddress set by an ADD ISDN CALL command on page 4-44. In this case the subaddress is checked twice, once by Q.931 and once at the ISDN call level.

The NUM1, NUM2, SUB1, SUB2, SPID1 and SPID2 parameters may all be specified without a value, to clear the current value for the respective parameter.

The T3xx parameters set timeout values for their respective timer, whether the timer is valid for a particular profile or not. Setting a timer to OFF disables the use of the timer.

Examples To use the 5ESS profile at 56K on interface BRI1, use the command:

```
SET Q931=bri1 PROFILE=5ess RATE=56k
```

See Also SET SYSTEM TERRITORY in *Chapter 1, Operation*
SHOW Q931

SHOW BRI CONFIGURATION

Syntax SHOW BRI [=instance] CONFIGURATION

where:

■ instance is the number of the BRI interface.

Description This command displays information about the modules which have been attached to the BRI interface (Figure 4-12 on page 4-87, Table 4-32 on page 4-87).

This example shows that the LAPD module is attached to the D channel and the PPP module to channel 0 which is using slot B1. The address referred to is the 16-bit field of the layer 2 frame which contains the SAPI and TEI for a D channel frame. The BRI hardware in the router is able to filter received frames based on a list of up to four addresses and an address mask. This reduces the loading on the BRI software module by not interrupting it for frames which are intended for other TEs.

The address mask indicates which bits of a frame's address field are significant when the comparison with each of the four addresses is made. A one bit in the address mask denotes a significant bit. As the B slots are not shared with other TEs the address filtering features of the hardware are not used for channels other than the D channel.

Figure 4-12: Example output from the SHOW BRI CONFIGURATION command.

```
Configuration for BRI instance 0:
```

```
    D Channel:
Module ..... LAPD
Address mask ... fdff
Addresses:
00ff
fcff

    Channel 0:    Slots: B1
Module ..... PPP
Rate ..... 64kbps
Address mask ... 0000
Addresses:
    none
```

Table 4-32: Parameters displayed in the output of the SHOW BRI CONFIGURATION command.

Parameter	Meaning
Channel	The channel to which the information applies.
Module	The module attached to the channel.
Rate	The bandwidth of the channel.
Address Mask	A mask used to determine which bits of a frame's address field are significant for filtering purposes.
Addresses	Addresses used for filtering incoming layer 2 frames. The frame's address field is ANDed with the address mask and then compared with this list of addresses. If a match is not found, the frame is ignored.

In the example shown above, the mask indicates that all bits of the address are significant except for the command/response field bit. The first address value shown corresponds to a SAPI of 0 (call control procedures) and a TEI of 127 (broadcast TEI), and the second address to a SAPI of 63 (layer 2 management procedures) and a TEI of 127.

If it becomes necessary for the D channel to accept frames with more than four addresses then the address mask is adjusted (fewer one bits) so that four addresses are sufficient to select all required frames.

Examples To display the configuration of BRI interface 0, use the command:

```
SHOW BRI=0 CONFIGURATION
```

See Also SHOW BRI COUNTERS
SHOW BRI STATE

SHOW BRI COUNTERS

Syntax `SHOW BRI [=instance] COUNTERS [= { INTERFACE | BRI }]`

where:

- *instance* is the number of the BRI interface.

Description This command displays the MIB counters associated with the BRI interface. If an interface is not specified, the MIB counters for all BRI interfaces are displayed. If a counter category is not specified, all categories are displayed.

The COUNTER parameter specifies the category of counters to display. The INTERFACE option displays counters from the interfaces table of the interfaces MIB relating to the BRI interface (Figure 4-13 on page 4-88, Table 4-33 on page 4-88). The BRI option displays counters from the enterprise MIB specific to a Basic Rate interface. The output is divided into sections, one for the BRI as a whole and one for each D, B1 and B2 channel. The output varies depending on whether the BRI interface is an S/T interface (Figure 4-14 on page 4-90, Table 4-34 on page 4-92) or a U interface (Figure 4-15 on page 4-91, Table 4-34 on page 4-92), and for U interfaces the type of controller used. The IOM counters shown in Figure 4-15 on page 4-91 relate to the operation of the IOM bus used for communication between the CPU and the PEB2091 controller, and apply only to U interfaces using the PEB2091 controller.

Figure 4-13: Example output from the SHOW BRI COUNTERS=INTERFACE command.

BRI instance 0:	522 seconds	Last change at:	0 seconds
Interface MIB Counters			
Receive:		Transmit:	
ifInOctets	91192	ifOutOctets	483455
ifInUcastPkts	0	ifOutUcastPkts	150
ifInNUcastPkts	0	ifOutNUcastPkts	0
ifInDiscards	0	ifOutDiscards	0
ifInErrors	0	ifOutErrors	0
ifInUnknownProtos	0	ifOutQLen	0

Table 4-33: Parameters displayed in the output of the SHOW BRI COUNTERS=INTERFACE command.

Parameter	Meaning
ifInOctets	The number of octets received on this interface.
ifInUcastPkts	The number of unicast frames delivered to a higher layer protocol.
ifInNUcastPkts	The number of non-unicast frames delivered to a higher-layer protocol.
ifInDiscards	The number of inbound frames discarded though no errors were detected to prevent them being delivered to higher-layer protocols.
ifInErrors	The number of inbound frames that contained errors preventing them being delivered to a higher-layer protocol.
ifInUnknownProtos	The number of frames which were discarded because they were for an unconfigured protocol.
ifOutOctets	The number of octets transmitted, including framing.

Table 4-33: Parameters displayed in the output of the SHOW BRI COUNTERS=INTERFACE command. (Continued)

Parameter	Meaning
ifOutUcastPkts	The number of unicast frames transmitted or discarded.
ifOutNUcastPkts	The number of non-unicast frames transmitted or discarded.
ifOutDiscards	The number of frames discarded though no errors had been detected preventing their being transmitted.
ifOutErrors	The number of frames not transmitted because of errors.
ifOutQLen	Length of output frame queue.

Figure 4-14: Example output from the SHOW BRI COUNTERS=BRI command for an S/T interface.

BRI instance 0:	2754 seconds	Last change at:	0 seconds
BRI Counters			
ActivationRequests	0	Activations	0
FramingViolations	0	UnbalancedFrames	0
Channel 0: Slots: B1			
Receive:		Transmit:	
Frames	0	Frames	0
OverlengthFrames	0	CTSLosts	0
UnderlengthFrames	0	Underruns	0
CRCErrors	0	LostInterrupts	0
Aborts	0	DroppedFrames	0
NonOctetAligneds	0	NoPackets	0
Overruns	0	QueueLength	0
NonmatchAddresses	0	Recovers	0
Misseds	0	SDMABusErrors	0
TooFewBuffers	0	CommandTimeouts	0
QueueLength	0	LastCommand	0
Channel 1: Slots: B2			
Receive:		Transmit:	
Frames	0	Frames	0
OverlengthFrames	0	CTSLosts	0
UnderlengthFrames	0	Underruns	0
CRCErrors	0	LostInterrupts	0
Aborts	0	DroppedFrames	0
NonOctetAligneds	0	NoPackets	0
Overruns	0	QueueLength	0
NonmatchAddresses	0	Recovers	0
Misseds	0	SDMABusErrors	0
TooFewBuffers	0	CommandTimeouts	0
QueueLength	0	LastCommand	0
D Channel:			
Receive:		Transmit:	
Frames	0	Frames	0
OverlengthFrames	0	CTSLosts	0
UnderlengthFrames	0	Underruns	0
CRCErrors	0	LostInterrupts	0
Aborts	0	DroppedFrames	0
NonOctetAligneds	0	NoPackets	0
Overruns	0	QueueLength	0
NonmatchAddresses	0	Recovers	0
Misseds	0	SDMABusErrors	0
TooFewBuffers	0	CommandTimeouts	0
QueueLength	0	LastCommand	0
Collisions	0	HighPriorityFrames	0

Figure 4-15: Example output from the SHOW BRI COUNTERS=BRI command for a U interface.

BRI instance 0:	2863 seconds	Last change at:	0 seconds
BRI Counters			
IntActivationRequests	0	ExtActivationRequests	0
Activations	0	ActivationFailures	0
TransparencyLosses	0	SynchronisationLosses	0
NetworkDeactivations	0	UnexpectedDeactivations	0
NearEndBlockErrors	0	FarEndBlockErrors	0
IOM Counters			
Receive:		Transmit:	
Bytes	0	Bytes	0
Messages	0	Messages	0
Overlength messages	0	Overlength messages	0
Zero length messages	0	Zero length messages	0
Protocol errors	0	Internal errors	0
Data buffer fulls	0	Not readys	0
Message buffer fulls	0	Aborts	0
		Excessive retries	0
Recovers	0	Timeouts	0
Channel 0: Slots: B1			
Receive:		Transmit:	
Frames	0	Frames	0
OverlengthFrames	0	CTSLosses	0
UnderlengthFrames	0	Underruns	0
CRCErrors	0	LostInterrupts	0
Aborts	0	DroppedFrames	0
NonOctetAligneds	0	NoPackets	0
Overruns	0	QueueLength	0
NonmatchAddresses	0	Recovers	0
Misseds	0	SDMABusErrors	0
TooFewBuffers	0	CommandTimeouts	0
QueueLength	0	LastCommand	0
Channel 1: Slots: B2			
Receive:		Transmit:	
Frames	0	Frames	0
OverlengthFrames	0	CTSLosses	0
UnderlengthFrames	0	Underruns	0
CRCErrors	0	LostInterrupts	0
Aborts	0	DroppedFrames	0
NonOctetAligneds	0	NoPackets	0
Overruns	0	QueueLength	0
NonmatchAddresses	0	Recovers	0
Misseds	0	SDMABusErrors	0
TooFewBuffers	0	CommandTimeouts	0
QueueLength	0	LastCommand	0
D Channel:			
Receive:		Transmit:	
Frames	0	Frames	0
OverlengthFrames	0	CTSLosses	0
UnderlengthFrames	0	Underruns	0
CRCErrors	0	LostInterrupts	0
Aborts	0	DroppedFrames	0
NonOctetAligneds	0	NoPackets	0
Overruns	0	QueueLength	0
NonmatchAddresses	0	Recovers	0
Misseds	0	SDMABusErrors	0
TooFewBuffers	0	CommandTimeouts	0
QueueLength	0	LastCommand	0
HighPriorityFrames	0		

Table 4-34: Parameters displayed in the output of the SHOW BRI COUNTERS=BRI command.

Parameter	Meaning
BRI instance	The instance number of the BRI interface.
seconds	The current value of <i>sysUpTime</i> .
Last change at	The value of <i>sysUpTime</i> at which the interface was last initialised.
BRI counters	Counters for the Basic Rate interfaces as a whole.
ActivationRequests	The number of valid activation requests.
FramingViolations	The number of framing violations seen by the transceiver.
Activations	The number of S/T or U loop activations.
UnbalancedFrames	The number of unbalanced frames seen by the transceiver.
IntActivationRequests	The number of internally generated activation requests.
TransparencyLosses	The number of times data transparency through the U interface controller has been lost.
NetworkDeactivations	The number of times the U loop has been deactivated by the network.
NearEndBlockErrors	The number of U loop frame CRC errors detected.
ExtActivationRequests	The number of network generated activation requests.
ActivationFailures	The number of times an activation attempt has failed.
SynchronisationLosses	The number of times synchronisation with the network over the U loop has been lost.
UnexpectedDeactivations	The number of times the U interface has been deactivated without being initiated by the network.
FarEndBlockErrors	The number of CRC errors in U loop frames transmitted by the router and reported by the network.
IOM Counters	Counters for the IOM controller. Displayed only for U interfaces using the PEB2091 controller.
Bytes	The number of bytes transmitted/received over the IOM bus.
Messages	The number of messages transmitted/received over the IOM bus.
Overlength messages	The number of overlength messages transmitted/received over the IOM bus.
Zero length messages	The number of messages zero length received over the IOM bus, or the number of zero length messages queued for transmission.
Protocol errors	The number of handshaking errors detected by the IOM bus controller while receiving a message.
Data buffer fulls	The number of times the buffer for characters received over the IOM bus has filled.
Message buffer fulls	The number of times the buffer for messages received over the IOM bus has filled.
Excessive retries	The number of times a message has been retransmitted 8 times without success.
Internal errors	The number of times an attempt was made to transmit an IOM bus message while a transmission was already in progress.

Table 4-34: Parameters displayed in the output of the SHOW BRI COUNTERS=BRI command. (Continued)

Parameter	Meaning
Not readys	The number of times an attempt was made to transmit an IOM bus message but the IOM bus controller was not ready.
Aborts	The number of times the PEB2091 requested that the transmission of a IOM bus message be aborted.
Timeouts	The number of times the transmission of an IOM bus message failed because of a timeout.
D Channel, Channel <i>n</i>	Counters for the D, B1 and B2 channels.
Slots	The slot used by the associated channel.
Frames	The number of frames received/transmitted.
OverlengthFrames	The number of overlength frames received.
UnderlengthFrames	The number of frames discarded because they were too short.
CRCErrors	The number of frames received with a CRC error.
Aborts	The number of received frames terminated with an abort.
NonoctetAligned	The number of non-octet aligned frames received.
Overruns	The number of frames lost due to a receive overrun.
NonmatchAddresses	The number of incoming frames rejected due to a non-matching address.
Misseds	The number of receive frames lost because lack of receive buffers.
TooFewBuffers	The number of received frames discarded because the number of buffers in the router had reached a critical level.
QueueLength	The length of the channel's receive/transmit queue.
Collisions	The number of times a frame had to be retransmitted on the D channel due to a collision.
CTSLosts	The number of frames during which the CTS input was negated.
Underruns	The number of times a frame had to be retransmitted due to a transmitter underrun.
LostInterrupts	The number of times the transmission or reception of a frame on the indicated channel had to be aborted due to no transmit/receive interrupt being received.
DroppedFrames	The number of frames discarded because the maximum transmit queue length was exceeded.
NoPackets	The number of times the 68302 or 68360 reported a transmit error, but there was no packet being transmitted or the packet in error could not be identified.
Recovers	The number of times the HDLC or IOM controller was reset due to a serious error or a RESET BRI command on page 4-72.
SDMABusErrors	The number of bus errors experienced by the HDLC controller.
CommandTimeouts	The number of times a command to the Ethernet hardware did not complete before the timeout timer expired.

Table 4-34: Parameters displayed in the output of the SHOW BRI COUNTERS=BRI command. (Continued)

Parameter	Meaning
LastCommand	The code of the command that was to be issued when a command timeout was detected.
HighPriorityFrames	The number of D channel high priority frames transmitted.

Examples To display the interface counters for BRI interface 0, use the command:

```
SHOW BRI=0 COUNTERS=INTERFACE
```

See Also RESET BRI COUNTERS
SHOW BRI CONFIGURATION

SHOW BRI CTEST

Syntax SHOW BRI[=*instance*] CTEST

where:

- *instance* is the number of the BRI interface.

Description This command displays the settings of the conformance test switches. If the interface is not specified, the settings for all BRI interfaces are displayed (Figure 4-16 on page 4-94, Table 4-35 on page 4-94).

Figure 4-16: Example output from the SHOW BRI CTEST command.

CTest switches for BRI instance 0:		
Number	Action	Status
1	Activation Request	no
2	Digital Loop (loopback 4)	no
3	B1, B2 channels transmit all zeroes	no
4	D channel transmit high priority frames ..	no
5	D channel transmit low priority frames ...	no
6	B1 channel transmit fox frames	no
7	B2 channel transmit fox frames	no
8	D channel transmit single zero frames	no

Table 4-35: Parameters displayed in the output of the SHOW BRI CTEST command.

Test	Function
1	An activation request is issued to the transceiver which will transmit INFO 1 in an attempt to activate the S/T loop. The status of the test is reset to "no" once the loop activates or when the activate timer times out. This conformance test has no effect if the loop is already activated.
2	Data received by the BRI module for both B channel and the D channel from the S/T loop is retransmitted on the same channel. This corresponds to loopback 4 defined in Appendix I of ITU-T Recommendation I.430.

Table 4-35: Parameters displayed in the output of the SHOW BRI CTEST command. (Continued)

Test	Function
3	HDLC frames containing all zeroes is transmitted continuously on both B channels.
4	High priority HDLC frames containing a fox message are transmitted on the D channel continuously.
5	Low priority HDLC frames containing a fox message are transmitted on the D channel continuously.
6	HDLC frames containing a fox message are transmitted on the B1 channel continuously.
7	HDLC frames containing a fox message are transmitted on the B2 channel continuously.
8	HDLC frames containing bytes with one zero and seven ones are transmitted on the D channel continuously.

Examples To display the conformance tests current running on BRI interface 0, use the command:

```
SHOW BRI=0 CTEST
```

See Also DISABLE BRI CTEST
ENABLE BRI CTEST
DISABLE BRI TEST
ENABLE BRI TEST
SHOW BRI TEST

SHOW BRI DEBUG

Syntax SHOW BRI [=instance] DEBUG

where:

- *instance* is the number of the BRI interface.

Description This command displays the settings of the debug switches. If the interface is not specified, the settings for all BRI interfaces are displayed (Figure 4-17 on page 4-95, Table 4-36 on page 4-96).

Figure 4-17: Example output from the SHOW BRI DEBUG command.

```
Debug switches for BRI instance 0:

Errors ..... no
Indications ..... no
State changes ... no
Events ..... no
```

Table 4-36: Parameters displayed in the output of the SHOW BRI DEBUG command.

Parameter	Meaning
Errors	A BRI software module internal error.
Indications	An indication from the layer 1 state machine to a higher layer or the management layer.
State changes	A change of state for the layer 1 state machine.
Events	An event that is an input to the layer 1 state machine.

Examples To display the state of debugging options for BRI interface 0, use the command:

```
SHOW BRI=0 DEBUG
```

See Also DISABLE BRI DEBUG
ENABLE BRI DEBUG

SHOW BRI STATE

Syntax SHOW BRI [=instance] STATE

where:

- *instance* is the number of the BRI interface.

Description This command displays information about the current state of the BRI interface. If the interface is not specified, the state of all BRI interfaces is displayed. The output varies depending on whether the BRI interface is an S/T interface (Figure 4-18 on page 4-96, Table 4-37 on page 4-97) or a U interface (Figure 4-19 on page 4-98, Table 4-39 on page 4-98).

Figure 4-18: Example output from the SHOW BRI STATE command for an S/T interface.

```
State for BRI instance 0:

Interface type ..... TE
State ..... Activated
Rx INFO ..... INFO 4
Tx INFO ..... INFO 3
Activate request ... no
Activated ..... yes
Synchronised ..... yes
Activation mode .... normal
Mode ..... mixed
ISDN slots ..... B1
TDM slots ..... B2
D channel class .... high
B1 enabled ..... yes
B2 enabled ..... no
B1, B2 aggregated... no
Rx multiframing .... no
Transceiver mask .. 55
```

Table 4-37: Parameters displayed in the output of the SHOW BRI STATE command for an S/T interface.

Parameter	Meaning
Interface type	The operational mode for the interface: TE or NT. The interface should only be configured as an NT for manufacturers testing.
State	The state of the physical layer state machine. See Table 4-38 on page 4-97 for a list of valid states.
Rx INFO	The INFO signals currently being received from the NT by the interface. In normal operation the BRI transceiver receives INFO 4.
Tx INFO	The INFO signals currently being transmitted to the NT by the interface. In normal operation the BRI transceiver transmits INFO 3.
Activate request	Whether or not an activation request has been received from a higher layer and is being processed.
Activated	Whether or not the loop is activated.
Synchronised	Whether or not the TE is synchronised to the NT.
Activation mode	The activation mode of the interface; one of "normal" or "always". The latter may be required for semipermanent connections.
Mode	The mode of the interface; one of "ISDN", "TDM" or "mixed".
ISDN slots	The list of slots reserved for ISDN calls. Only valid when the interface is not in TDM mode.
TDM slots	The list of slots reserved for TDM groups. Only valid when the interface is not in ISDN mode.
D channel class	The current D channel priority class. This may vary from one D channel frame to the next.
B1/B2 enabled	Whether or not the B channels are attached to a higher layer module.
B1, B2 aggregated	Whether or not the B channels are aggregated.
Rx multiframing	Whether or not the transceiver has detected multiframing in the data stream from the NT. The router does not currently support multiframing.
Transceiver mask	The mask revision of the S/T transceiver chip (for some hardware models only).

Table 4-38: States of the physical layer state machine for an ISDN Basic Rate S/T Interface.

State	Meaning
Inactive	Power has not been applied to the interface. This state should never be seen.
Sensing	The initial state at power-on, before the S/T transceiver has determined what signal it is receiving.
Deactivated	The transceiver is receiving INFO 0 from the NT.

Table 4-38: States of the physical layer state machine for an ISDN Basic Rate S/T Interface. (Continued)

State	Meaning
Awaiting Signal	A transitory state entered when the transceiver has been given an activation request.
Identifying Input	This state is entered from Awaiting Signal when the transceiver has detected a signal but has not yet determined which INFO signal it is.
Synchronized	This state is entered when the transceiver is receiving INFO 2 from the NT, i.e. it has synchronised to the NT.
Activated	This is the normal operational state. The transceiver is receiving INFO 4 from the NT.
Lost framing	This state is entered if the transceiver loses synchronisation with the signal transmitted by the NT.

Figure 4-19: Example output from the SHOW BRI STATE command for a U interface.

```

State for BRI instance 0:

Interface type ..... TE
State ..... Active
Activate request ... no
Activated ..... yes
Synchronised ..... yes
Transparent ..... yes
Activation mode .... normal
EOC message ..... broadcast command - return to normal
Maintenance mode ... none
Mode ..... ISDN
ISDN slots ..... B1, B2
B1 enabled ..... no
B2 enabled ..... no
B1, B2 aggregated... no
Transceiver mask .. 03

```

Table 4-39: Parameters displayed in the output of the SHOW BRI STATE command for a U interface.

Parameter	Meaning
Interface type	The operational mode for the interface; one of "TE" or "LT". The LT option appears only for a special test mode of some hardware models.
State	The state of the physical layer state machine. See Table 4-40 on page 4-99 for a list of valid states.
Activate request	Whether or not an activation request has been received from a higher layer and is being processed; one of "yes" or "no".
Activated	Whether or not the loop is activated; one of "yes" or "no".
Synchronised	Whether or not the TE is synchronised to the LT; one of "yes" or "no".
Transparent	Whether or not the U interface transceiver is passing data between the router and the network; one of "yes" or "no".

Table 4-39: Parameters displayed in the output of the SHOW BRI STATE command for a U interface. (Continued)

Parameter	Meaning
Activation mode	The activation mode of the interface; always "normal".
EOC message	The message most recently received over the Embedded Operations Channel from the network.
Maintenance mode	The maintenance mode of the interface; one of "none", "Quiet", or "Insertion Loss Test Mode".
Mode	The mode of the interface; always "ISDN" (TDM mode is not available on U interfaces).
ISDN slots	The list of slots reserved for ISDN calls.
B1/B2 enabled	Whether or not the B channels are attached to a higher layer module; one of "yes" or "no".
B1, B2 aggregated	Whether or not the B channels are aggregated; one of "yes" or "no".
Transceiver mask	The mask revision of the U transceiver chip (for some hardware models only).

Table 4-40: States of the physical layer state machine for an ISDN Basic Rate U Interface.

State	Meaning
Deactivated	The U loop is idle, no signals are being received or transmitted.
Activating	The U loop is in the process of activation, this may take up to 15 seconds.
Pending active	The router and the LT have synchronised to one another, the router is waiting to receive "act"=1 from the LT.
Active	The U loop is active, the normal operational state.
Pending deactivated	The router has received "dea"=0 from the LT and is waiting for the U loop to be completely deactivated (no signal received).

Examples To display information about the current state of BRI interface 0, use the command:

```
SHOW BRI=0 STATE
```

See Also SHOW BRI CONFIGURATION
SHOW BRI COUNTERS

SHOW BRI TEST

Syntax SHOW BRI [=instance] TEST

where:

- *instance* is the number of the BRI interface.

Description This command displays the settings of the test switches. If the interface is not specified, the settings for all BRI interfaces are displayed. The output varies depending on whether the BRI interface uses a MC145474 controller (Figure 4-20 on page 4-100, Table 4-41 on page 4-100), a PSB2186 controller (Figure 4-21 on page 4-101, Table 4-42 on page 4-101), a PEB2091 controller (Figure 4-22 on page 4-102, Table 4-43 on page 4-102) or a MC145572 controller (Figure 4-23 on page 4-102, Table 4-44 on page 4-103).

Figure 4-20: Example output from the SHOW BRI TEST command for BRI interfaces using an MC145474 controller.

Test switches for BRI instance 0:		
Number	Action	Status
1	IMP IDL Loop	no
2	IMP IDL Echo	no
3	Transceiver 2B+D IDL Non-Transp Loop	no
4	Transceiver B1 IDL Non-Transp Loop	no
5	Transceiver B2 IDL Non-Transp Loop	no
6	Transceiver B1 S/T Transp Loop	no
7	Transceiver B2 S/T Transp Loop	no
8	Transceiver B1 S/T Non-Transp Loop	no
9	Transceiver B2 S/T Non-Transp Loop	no
10	Transceiver External S/T Loop	no
11	Transceiver 96kHz Test Tone	no
12	Transceiver Activation Proc Disable	no
13	Transceiver D Channel Proc Ignore (TE) ...	no
14	Transceiver Map E Bits to IDL (TE)	no
15	Transceiver IDL Free Run (TE)	no

Table 4-41: ISDN Basic Rate Interface test modes for S/T interfaces using an MC145474 controller.

Test	Function
1	A loopback by the IMP of the data on the IDL bus towards the IMP.
2	A loopback by the IMP of the data on the IDL bus towards the interface.
3	A loopback by the transceiver of the B and D channel data on the IDL bus towards the IMP. Idles are transmitted on to the S/T loop.
4	A loopback by the transceiver of the B1 channel data on the IDL bus towards the IMP. Idles are transmitted on to the S/T loop in place of B1 data.
5	A loopback by the transceiver of the B2 channel data on the IDL bus towards the IMP. Idles are transmitted on to the S/T loop in place of B2 data.
6	A loopback by the transceiver of the B1 channel data on the S/T loop towards the S/T loop. The data is also passed through to the IDL bus, but data received on the IDL bus for channel B1 is ignored.

Table 4-41: ISDN Basic Rate Interface test modes for S/T interfaces using an MC145474 controller. (Continued)

Test	Function
7	A loopback by the transceiver of the B2 channel data on the S/T loop towards the S/T loop. The data is also passed through to the IDL bus. Data received on the IDL bus for channel B2 is ignored.
8	A loopback by the transceiver of the B1 channel data on the S/T loop towards the S/T loop. The data is not passed through to the IDL bus, idles are transmitted in its place. Data received on the IDL bus for channel B1 is ignored.
9	A loopback by the transceiver of the B2 channel data on the S/T loop towards the S/T loop. The data is not passed through to the IDL bus, idles are transmitted in its place. Data received on the IDL bus for channel B2 is ignored.
10	The transceiver will receive and demodulate its own transmitted data provided the transmit pair is connected to the receive pair at the interface connector. For this test to work correctly tests 12 and 15 should also be enabled.
11	A 96kHz test tone is transmitted on to the S/T loop.
12	The transceiver is forced into the highest INFO state, i.e. the transceiver transmits INFO 4 for a TE or INFO 3 for a NT.
13	The transceiver transmits without regard for the D channel contention procedures governing transmission. This test is applicable to a TE only.
14	The transceiver outputs E channel data on to the IDL bus in place of the D channel data received from the NT. This test is applicable to a TE only.
15	The transceiver will clock the IDL bus even if it is not able to derive a clock from the S/T loop. This test is applicable to a TE only.

Figure 4-21: Example output from the SHOW BRI TEST command for BRI interfaces using a PSB2186 controller.

Test switches for BRI instance 0:		
Number	Action	Status
1	Send Single Zeroes	no
2	Send Continuous Zeroes	no
3	Test Loop 3 (internal)	no
4	Test Loop 4 (external)	no

Table 4-42: ISDN Basic Rate Interface test modes for S/T interfaces using a PSB2186 controller.

Test	Function
1	Single alternating pulses are sent at a 2kHz repetition rate.
2	Continuous alternating pulses are sent.
3	Data transmitted by the router is internally looped back to its receiver.
4	Data received at the interface is looped back out of the interface by the transceiver.

Figure 4-22: Example output from the SHOW BRI TEST command for BRI interfaces using a PEB2091 controller.

```

Test switches for BRI instance 0:

```

Number	Action	Status
1	Force reset	no
2	Force SN3	no
3	Enable analogue loopback	no
4	Enable 2B+D test access port	no
5	B1 loopback	no
6	B2 loopback	no
7	2B+D loopback	no
8	Activated LED on tx SN3	no

Table 4-43: ISDN Basic Rate Interface test modes for U interfaces using a PEB2091 controller.

Test	Function
1	Force a reset of the controller so that it enters quiet mode and does not transmit on the U loop.
2	Force the controller to transmit SN3 (standard framed, scrambled signal) on the U loop.
3	Enable an analogue loopback so that the router receives the data it transmits.
4	Enable the internal 2B + D test access port.
5	Enable a loopback of the B1 channel data on the U loop towards the U loop.
6	Enable a loopback of the B2 channel data on the U loop towards the U loop.
7	Enable a loopback of the B1, B2 and D channel data on the U loop towards the U loop.
8	Turn the activated LED on as soon as SN3 is transmitted to the LT, rather than when "act"=1 is received.

Figure 4-23: Example output from the SHOW BRI TEST command for BRI interfaces using a MC145572 controller.

```

Test switches for BRI instance 0:

```

Number	Action	Status
1	Force reset	no
2	Force SN3	no
3	Enable analogue loopback	no
4	Enable 2B+D test access port	no
5	B1 both direction loopbacks	no
6	B2 both direction loopbacks	no
7	2B+D both direction loopbacks	no
8	Activated LED on tx SN3	no
9	Simulate LT mode	no

Table 4-44: ISDN Basic Rate Interface test modes for U interfaces using an MC145572 controller.

Test	Function
1	Force a reset of the controller so that it enters quiet mode and does not transmit on the U loop.
2	Force the controller to transmit SN3 (standard framed, scrambled signal) on the U loop.
3	Enable an analogue loopback so that the router receives the data it transmits.
4	Enable the internal 2B + D test access port.
5	Enable a loopback of the B1 channel data on the U loop towards the U loop and data transmitted by the router back to the router.
6	Enable a loopback of the B2 channel data on the U loop towards the U loop and data transmitted by the router back to the router.
7	Enable a loopback of the B1, B2 and D channel data on the U loop towards the U loop and data transmitted by the router back to the router.
8	Turn the activated LED on as soon as SN3 is transmitted to the LT, rather than when "act"=1 is received.
9	The interface will act as if it is an LT.

Examples To display the tests running on BRI interface 0, use the command:

```
SHOW BRI=0 TEST
```

See Also DISABLE BRI CTEST
ENABLE BRI CTEST
DISABLE BRI TEST
ENABLE BRI TEST
SHOW BRI CTEST

SHOW ISDN CALL

Syntax `SHOW ISDN CALL [= { acnum | name }]`

where:

- *acnum* is the index of an active ISDN call.
- *name* is an ISDN call name, 1 to 15 characters in length. Valid characters are letters (a–z, A–Z), decimal digits (0–9) and underscore (“_”). It is case-insensitive.

Description This command displays information about ISDN call definitions and active calls. If an active call number or call name is not specified, summary details of all ISDN call definitions and active calls are displayed (Figure 4-24 on page 4-104, Table 4-45 on page 4-104). If an active call number or call name is specified, detailed information about the particular active call or call definition is displayed (Figure 4-25 on page 4-105, Table 4-46 on page 4-105).

Figure 4-24: Example output from the SHOW ISDN CALL command.

ISDN call details					
Name	Number	Remote call	State	Precedence	
HeadOffice	3432114	Regional	(E) IN & OUT	IN	
ISDN active calls					
Index	Name	Interface	User	State	Prec
0	HeadOffice	BRI0	03-00	ON	No

Table 4-45: Parameters displayed in the output of the SHOW ISDN CALL command.

Parameter	Meaning
ISDN call details	Information about all defined ISDN calls.
Name	The name of the ISDN call.
Number	The number to call.
Remote call	The remote call name.
State	The state of this call definition; one of "(E)" (enabled) or "(D)" (disabled), and the directions for which the call is enabled.
Precedence	For call definitions, the direction of precedence of the call.
ISDN active calls	Information about active ISDN calls.
Index	A number identifying an active ISDN call.
Interface	The ISDN interface used for the call.
User	The module and instance of the higher layer module using the call.
State	The state of the active call; one of "ON", "TRY" or "WAIT".
Prec	For active calls, whether or not this call actually has precedence.

Figure 4-25: Example output from the SHOW ISDN CALL command for a specified call name.

```

Call name ..... HeadOffice
Enabled ..... Yes
Remote call ..... Regional
Called number ..... 3432114
Calling number ..... -
Calling subaddress ..... -
Direction ..... IN & OUT
Precedence ..... IN
Required interface ..... NONE
Preferred interface ..... NONE
Data rate ..... 64k
Use Data Over Voice ..... No
Priority ..... 50
Bump delay ..... 5
Holdup time ..... 0s
Keep call up ..... No
Call back ..... No
Call back delay ..... 41
RN1 (retries per group) ... 0
RT1 (between retries) ..... 30s
RN2 (retry groups) ..... 0
RT2 (between groups) ..... 600s
Alternate number ..... -
Out called subaddress ..... Remote name
Out user data ..... -
Out CLI ..... -
In called sub search ..... Local name
In called sub check ..... -
In user data search ..... -
In user data check ..... -
In CLI search ..... No
In CLI check ..... -
In CLI list ..... none
Match any call ..... No
User type ..... ATTACH
  PPP template ..... Default
Login type ..... User
Login user name ..... none
Login password ..... none

Number of attachments ..... 1
User module ..... PPP
Attachment ..... 0

```

Table 4-46: Parameters displayed in the output of the SHOW ISDN CALL command for a specified call name.

Parameter	Meaning
Call name	The name of the ISDN call.
Enabled	Whether or not the call is enabled; one of "Yes" or "No".
Remote call	The remote call for this call.
Called number	The number called for this call.
Calling number	The number called from for this call.
Calling subaddress	The subaddress called from for this call.
Direction	The directions for which the call is enabled.

Table 4-46: Parameters displayed in the output of the SHOW ISDN CALL command for a specified call name. (Continued)

Parameter	Meaning
Precedence	The direction of precedence for this call.
Required interface	The required interface for this call.
Preferred interface	The preferred interface for this call.
Data rate	The data rate to use when making an outgoing call with this call; one of "56K" or "64K".
Use Data Over Voice	Whether or not the call setup message for this call will specify voice bearer capability or data bearer capability; one of "Yes" (voice bearer) or "No" (data bearer).
Priority	The priority of this call.
Bump delay	The delay, in tenths of a second, between a call being initiated and the required Q.931 response, if another call must be bumped to allow this call to proceed.
Holdup time	The minimum time, in seconds, that the call will be held up before being dropped.
Keep call up	Whether or not this call is to be kept up always.
Call back	Whether or not the router should hang up the incoming call and call back when this call is selected.
Call back delay	The delay, in tenths of a second, before calling back, if call back is enabled.
RN1	The number of retries in a retry group.
RT1	The time, in seconds, between retries in a retry group.
RN2	The number of time the retry group is repeated.
RT2	The time, in seconds, between repeats of the retry group.
Alternate number	The alternate number dialed when retries have failed.
Out called subaddress	The format of the called party subaddress IE in the outgoing SETUP message. This is set to the call's name or remote call name with the OUTSUB parameter, or set to an arbitrary sequence of digits with the SUBADDRESS parameter.
Out user data	The format of the user-user data IE in the outgoing SETUP message.
Out CLI	The format of the calling party number IE (CLI) in the outgoing SETUP message.
In called sub search	How to use the called party subaddress IE in incoming SETUP messages in searching for this call.
In called sub check	How to use the called party subaddress IE in incoming SETUP messages in checking this call.
In user data search	How to use the user-user data IE in incoming SETUP messages in searching for this call.
In user data check	How to use the user-user data IE in incoming SETUP messages in checking this call.
In CLI search	Whether or not to use the CLI in incoming SETUP messages to search for this call.
In CLI check	How to use the CLI in incoming SETUP messages to check this call.

Table 4-46: Parameters displayed in the output of the SHOW ISDN CALL command for a specified call name. (Continued)

Parameter	Meaning
In CLI list	The index of the CLI list to use, if required, for checking CLI against this call.
Match any call	Whether or not this call can be used to answer any incoming call, if no other call has already been found to match the incoming call.
User type	The way that users attach to this ISDN call. One of "ATTACH" (users attach explicitly) or "PPP" (a dynamic PPP interface is created).
PPP template	The PPP template to use when creating a dynamic PPP interface, or "Default" if the default PPP template is used.
Login type	The method of authentication for incoming calls.
Login username	The source of the username in login procedures.
Login password	The source of the password in login procedures.
Number of attachments	The number of attachments from higher layer modules for this call.
User module	The higher layer module that is attached to this call.
Attachment	The instance number (for the higher layer module) for this attachment. This line may be repeated.

Examples To display the configuration of ISDN call "Region-1", use the command:

```
SHOW ISDN CALL="Region-1"
```

See Also ACTIVATE ISDN CALL
ADD ISDN CALL
DEACTIVATE ISDN CALL
DELETE ISDN CALL
DISABLE ISDN CALL
ENABLE ISDN CALL
SET ISDN CALL

SHOW ISDN CLILIST

Syntax SHOW ISDN CLILIST[=0..99]

Description This command displays a specified CLI list or all CLI lists, if no list is specified (Figure 4-26 on page 4-108, Table 4-47 on page 4-108). The numbers in CLI lists are ordered as they are added to the list, a fact reflected in the display of the list.

Figure 4-26: Example output from the SHOW ISDN CLILIST command.

```
ISDN CLI list 0
Total fails: 5
Number                               Matches
-----
045660234                             12
3432115                               1
-----

ISDN CLI list 1
Total fails: 104
Number                               Matches
-----
3430803                             124
3430804                             59
-----
```

Table 4-47: Parameters displayed in the output of the SHOW ISDN CLILIST command.

Parameter	Meaning
ISDN CLI list	The index of the ISDN CLI list being displayed.
Total fails	The number of times a number being checked against this list was not matched by any number in the list.
Number	The ISDN phone number for this entry in the CLI list.
Matches	The number of times a number being checked against this CLI list matched this number.

Examples To display all CLI lists, use the command:

```
SHOW ISDN CLILIST
```

See Also ADD ISDN CLILIST
DELETE ISDN CLILIST

SHOW ISDN DOMAINNAME

Syntax SHOW ISDN DOMAINNAME

Description This command displays the domain name to be used for ISDN DNS lookups. Only one ISDN domain name may be defined (Figure 4-27 on page 4-108).

Figure 4-27: Example output from the SHOW ISDN DOMAINNAME command.

```
The ISDN default domain name is: sales.southern.com
```

See Also ADD ISDN DOMAINNAME
DELETE ISDN DOMAINNAME
SET ISDN DOMAINNAME

SHOW ISDN LOG

Syntax SHOW ISDN LOG

Description This command displays the current contents of the call log (Figure 4-28 on page 4-109, Table 4-48 on page 4-109). The call logging facility records details of events associated with ISDN calls. Log entries are sorted according to the time the call was initiated. An entry is added to the log when a call is initiated. When the log exceeds a predefined maximum length, the oldest entry that is in the CLEARED state is removed from the log. If no entries qualify the log is allowed to grow larger than the maximum defined length. Log messages can be sent to an asynchronous port on the router when the log entry enters the CLEARED state.

The Q.931 cause code displayed in the *Cause* field of the output is returned by the ISDN network to the router each time a call is cleared, and can be used in debugging ISDN interconnection problems. See “ISDN Q.931 Call Clearance Cause Codes” on page B-6 of *Appendix B, Reference Tables* for a list of cause codes and their meanings for Q.931 call control profiles currently supported by the router. Not all cause codes are supported by all ISDN service providers.

Figure 4-28: Example output from the SHOW ISDN LOG command.

Call Name	Start Time	Duration	Dir	Number	Cause
HeadOffice	02-Mar-1995 17:46:38	CLEARED	OUT	3432114	N34,-
HeadOffice	02-Mar-1995 17:46:38	CLEARED	OUT	3432114	N34,-
HeadOffice	02-Mar-1995 17:46:38	CLEARED	IN		U88,113
HeadOffice	02-Mar-1995 17:46:38	CLEARED	IN		U88,113
HeadOffice	02-Mar-1995 17:48:22	0:03:25	OUT	3432114	U16,-
HeadOffice	02-Mar-1995 17:55:18	0:05:06	IN		U16,-
HeadOffice	02-Mar-1995 17:55:18	0:05:06	IN		U16,-
HeadOffice	02-Mar-1995 18:02:08	0:01:13	IN		U16,-
HeadOffice	02-Mar-1995 18:02:08	0:01:13	IN		U16,-
HeadOffice	02-Mar-1995 18:16:56	0:01:49	OUT	3432114	U16,-
HeadOffice	02-Mar-1995 18:16:56	0:01:49	OUT	3432114	U16,-
HeadOffice	03-Mar-1995 08:55:54	0:03:30	OUT	3432114	U16,-
HeadOffice	03-Mar-1995 08:55:54	0:03:30	OUT	3432114	U16,-
No ISDN logging port defined.					

Table 4-48: Parameters displayed in the output of the SHOW ISDN LOG command.

Parameter	Meaning
Call Name	The name of the call.
Start Time	The date and time the call was initiated.
Duration	The length of the call for a call that has been completed, or one of “INITIAL” (the call is being set up), “ACTIVE” (the call is still active), “DISCONNECT” (the call is being disconnected) or “CLEARED” (the call was cleared before becoming active).
Dir	The direction of the call; one of “OUT” or “IN”.
Number	The number being called.
Cause	The reason the call was disconnected. The first character is a “U” (disconnected by user) or an “N” (disconnected by network), followed by the Q.931 cause code and (for some causes) Q.931 diagnostic code.

Examples To display the ISDN call log, use the command:

```
SHOW ISDN LOG
```

See Also DISABLE ISDN LOG
DISABLE Q931 DEBUG
ENABLE ISDN LOG
ENABLE Q931 DEBUG
SET ISDN LOG

SHOW LAPD

Syntax SHOW LAPD [=interface]

where:

■ *interface* is a slotted interface number (0, 1, 2,...).

Description This command displays general information about LAPD interfaces (Figure 4-29 on page 4-110 and Figure 4-30 on page 4-111, Table 4-49 on page 4-111).

Figure 4-29: Example output from the SHOW LAPD command for a Basic Rate Interface.

```
Interfaces:
ISDN      Type      TEI Mode      Debug      TEI      NAS mode      NAS master
-----
BRI0      TE       automatic    off        066      Normal        -
              064

SAPs:
ISDN      SAPI      T200      T201      T202      T203      N200      N201      N202      k
-----
BRI0      063      000010  000010  000020  000100  000003  000260  000003  001
              000      000010  000010  000020  000100  000003  000260  000003  001

DLCs:
ISDN      SAPI      CES      TEI      State      V(S)      V(A)  rxN(S)      V(R)  rxN(R)
-----
BRI0      063      000      127      bcast      -          -          -          -          -
              000      000      127      bcast      -          -          -          -          -
              001      066      ALIVE     0038      0038      0002      0003      0038
              002      064      ALIVE     0039      0039      0000      0001      0039

Packet parameters:
-----
BRI0
  Packet mode TEIs:  1
  Packet mode SPIDs: -
-----
```


Figure 4-30: Example output from the SHOW LAPD command for a Primary Rate Interface.

Interfaces:									
ISDN	Type	TEI Mode		Debug	TEI	NAS mode		NAS master	
PRI0	TE	nonAuto		off	000	Normal		-	
PRI0	TE	nonAuto		off	000	Normal		-	
SAPs:									
ISDN	SAPI	T200	T201	T202	T203	N200	N201	N202	k
PRI0	063	000010	-	-	000100	000003	000260	-	007
	000	000010	-	-	000100	000003	000260	-	007
PRI1	063	000010	-	-	000100	000003	000260	-	007
	000	000010	-	-	000100	000003	000260	-	007
DLCs:									
ISDN	SAPI	CES	TEI	State	V(S)	V(A)	rxN(S)	V(R)	rxN(R)
PRI0	063	000	127	bcast	-	-	-	-	-
	000	000	000	ALIVE	0021	0021	0076	0077	0021
PRI0	063	000	127	bcast	-	-	-	-	-
	000	000	000	ALIVE	0014	0014	0051	0052	0014
Packet parameters:									
PRI0									
Packet mode TEIs: -									
Packet mode SPIDs: -									
PRI1									
Packet mode TEIs: -									
Packet mode SPIDs: -									

Table 4-49: Parameters displayed in the output of the SHOW LAPD command.

Parameter	Meaning
ISDN	The name of the ISDN interface.
Type	The operating mode of the interface; one of "TE" or "NT". The normal operating mode is TE, so NT should not appear.
TEI Mode	The TEI assignment mode; one of "Automatic" or "nonAuto".
Debug	The state of debugging; one of "off", "state", "pkt" or "st+pkt".
TEI	The Terminal Endpoint Identifier.
NAS mode	Non-associated signalling mode or common D channel mode. One of "Normal" (this interface's D channel does the signalling for this interface and no other interface), "Master" (this interface's D channel does the signalling for this interface and other interfaces) or "Slave" (another interface's D channel does the signalling for this interface).
NAS master	Non-associated signalling or common D channel master interface. If this interface's NAS mode is "Slave", the NAS master gives the interface whose D channel will provide the signalling channel for this interface. If the NAS mode is "Normal" or "Master", this field contains "-".
SAPI	The Service Access Point Identifier.
T20x	The value of timer T20x (in tenths of a second).
N20x	The value of counter N20x.

**Table 4-49: Parameters displayed in the output of the SHOW LAPD command.
(Continued)**

Parameter	Meaning
k	The value for K.
CES	The Connection Endpoint Suffix.
State	The state of the DLC; one of "ALIVE", "DEAD" or "bcast". bcast links have only a single state. For other links the state is ALIVE if the link can be used by higher protocol layers, or DEAD if it can not be used by higher protocol layers.
V(S)	The value of the internal V(S) count.
V(A)	The value of the internal V(A) count.
rxN(S)	The Number Sent count in the last received packet.
V(R)	The value of the internal V(R) count.
rxN(R)	The Number Received count in the last received packet.
Packet parameters	Parameters for X.25 packet mode operation
Packet mode TEIs	TEIs that have been configured for the use of X.25 over LAPD.
Packet mode SPIDs	Indices of SPIDs that are available for use by X.25 over LAPD.

Examples To display the configuration of LAPD interface 0, use the command:

```
SHOW LAPD=0
```

See Also SHOW LAPD COUNT
SHOW LAPD STATE

SHOW LAPD COUNT

Syntax SHOW LAPD [=interface] COUNT

where:

- *interface* is a slotted interface number (0, 1, 2,...).

Description This command displays the LAPD MIB counters for the ISDN interface and for each DLC of each SAP. If the interface is not specified, the MIB counters for all ISDN interfaces are displayed (Figure 4-31 on page 4-113, Table 4-50 on page 4-113).

Figure 4-31: Example output from the SHOW LAPD COUNT command.

ISDN	BRI0		
Total Receive		Total Transmit	
InOctets:	0000091114	OutOctets:	0000483389
InUcastPkts:	0000000000	OutUcastPkts:	0000000150
InNUcastPkts:	0000000000	OutNUcastPkts:	0000000000
InDiscards:	0000000000	OutDiscards:	0000000000
InErrors:	0000000000	OutErrors:	0000000000
InUnknownProtos:	0000000000		
ISDN	BRI0		
SAPI	063		
CES	000		
Receive		Transmit	
I Frames:	0000000000	I Frames:	0000000000
UI Frames:	0000000002	UI Frames:	0000000001
RR Frames:	0000000000	RR Frames:	0000000000
RNR Frames:	0000000000	RNR Frames:	0000000000
REJ Frames:	0000000000	REJ Frames:	0000000000
SABME Frames:	0000000000	SABME Frames:	0000000000
DM Frames:	0000000000	DM Frames:	0000000000
DISC Frames:	0000000000	DISC Frames:	0000000000
UA Frames:	0000000000	UA Frames:	0000000000
FRMR Frames:	0000000000	FRMR Frames:	0000000000
XID Frames:	0000000000	XID Frames:	0000000000
Errors			
?:	00000000	A:	00000000
D:	00000000	B:	00000000
H:	00000000	C:	00000000
L:	00000000	E:	00000000
		F:	00000000
		G:	00000000
		I:	00000000
		J:	00000000
		K:	00000000
		M:	00000000
		N:	00000000
		O:	00000000

Table 4-50: Parameters displayed in the output of the SHOW LAPD COUNT command.

Parameter	Meaning
ISDN	The name of the ISDN interface.
SAPI	The Service Access Point Identifier.
CES	The Connection Endpoint Suffix.
Total Receive	Number of frames received by the LAPD interface.
Total Transmit	Number of frames transmitted by the LAPD interface.
Receive	Number of frames received by the DLC.
Transmit	Number of frames transmitted by the DLC.
Errors	The number of times each error type has occurred.

Examples To display the counters for LAPD interface 0, use the command:

```
SHOW LAPD=0 COUNT
```

See Also SHOW LAPD
SHOW LAPD STATE

SHOW LAPD STATE

Syntax SHOW LAPD [=interface] STATE

where:

- *interface* is a slotted interface number (0, 1, 2,...).

Description This command displays the current and previous state of each DLC on the ISDN interface (Figure 4-32 on page 4-114, Table 4-51 on page 4-114).

Figure 4-32: Example output from the SHOW LAPD STATE command.

```
lapdCount 25045

ISDN  SAPI  CES  TEI  state - oldState
-----
BRI0   063   000  127  bcast(1 - 1)
      000   000  127  bcast(1 - 1)
      001   064  LAPD_ESTABLISHED(7) - LAPD_TIMER_RECOV(8)
-----
```

Table 4-51: Parameters displayed in the output of the SHOW LAPD STATE command.

Parameter	Meaning
ISDN	The name of the ISDN interface.
SAPI	The Service Access Point Identifier.
CES	The Connection Endpoint Suffix.
TEI	The Terminal Endpoint Identifier.
State	The current state of the DLC state machine.
oldState	The previous state of the DLC state machine.

Examples To display state information for LAPD interface 0, use the command:

```
SHOW LAPD=0 STATE
```

See Also SHOW LAPD
SHOW LAPD COUNT

SHOW Q931

Syntax SHOW Q931 [=interface] [CALL [=q931-call]]

where:

- *interface* is a slotted interface name or number. Interface names are formed by concatenating an interface type and instance (e.g. BRI0 or PRI1). Interface numbers are the decimal index of the slotted interface (0, 1, 2,...).
- *q931-call* is the number of a Q.931 call.

Description This command displays Q.931 profile and timer values, or active call information, for the specified ISDN interface.

If the CALL parameter is not specified, information about the Q.931 interface is displayed (Figure 4-33 on page 4-115, Table 4-52 on page 4-116). If the interface is not specified, the information is displayed for all ISDN interfaces.

If the CALL parameter is specified without a value, information about all Q.931 calls is displayed (Figure 4-34 on page 4-117, Table 4-53 on page 4-117). If the CALL parameter is specified with a value, information about the specified Q.931 call is displayed.

Figure 4-33: Example output from the SHOW Q931 command.

```
Q.931 interface ... BRI0
Profile ..... NI1-BR
ASD state ..... Operational
Data rate ..... 64k
Number 1 ..... -
Sub-address 1 ..... -
Number 2 ..... -
Sub-address 2 ..... -
DOV number ..... -
No number ..... Accept
No sub-address .... Accept
DLC1
  State ..... Established
  SPID state ..... OP
  SPID file state ... 3 (Auto SPID successful)
  Current SPID ..... 62155542310101
  USID ..... 0
  Terminal ID ..... 1
DLC2
  State ..... Established
  SPID state ..... OP
  SPID file state ... 3 (Auto SPID successful)
  Current SPID ..... 62155579340101
  USID ..... 0
  Terminal ID ..... 2
Common D channel
  Interface ID .... 00
TSPID ..... 20
T301 ..... -
T302 ..... -
T303 ..... 4
T304 ..... 15
T305 ..... 30
T308 ..... 4
T309 ..... 90
T310 ..... -
T313 ..... 4
T314 ..... -
T316 ..... -
T317 ..... -
T318 ..... -
T319 ..... -
T321 ..... -
T322 ..... 4
```

Table 4-52: Parameters displayed in the output of the SHOW Q931 command.

Parameter	Meaning
Q.931 interface	The ISDN interface.
Profile	The Q.931 profile in use on the interface; one of: "5ESS-BR" Lucent 5ESS custom (USA & Canada) Basic Rate "AUS-BR" Australian Telecom Basic Rate "AUS-PR" Australian Telecom Primary Rate "China-BR" China Telecom Basic Rate "China-PR" China Telecom Primary Rate "DMS100-BR" NorTel DMS-100 custom (USA & Canada) Basic Rate "ETS-BR" EU/EFTA countries ETSI Basic Rate. "ETS-PR" EU/EFTA countries ETSI Primary Rate. "JPN-BR" Japan Basic Rate. "JPN-PR" Japan Primary Rate. "KOREA-BR" Korea Basic Rate "KOREA-PR" Korea Primary Rate "NI1-BR" National ISDN (USA & Canada) Basic Rate "NZL-BR" New Zealand Telecom Basic Rate. "NZL-PR" New Zealand Telecom Primary Rate. "US ASD-BR" Auto switch detection (USA & Canada) Basic Rate
	The state of the auto switch detection state machine; one of "ASD-0", "ASD-1", "ASD-2", "ASD-3", "ASD-4", "ASD-5", "ASD-6" or "Operational".
Data rate	The data rate for this interface; one of "56k" or "64k".
Number 1, 2	The ISDN numbers assigned to the interface.
Sub-address 1, 2	The ISDN subaddresses assigned to the interface.
DOV number	The ISDN number assigned for DOV (Data Over Voice) calls. Voice calls received on this number will be treated as data calls, not voice calls.
No number	Whether to accept or reject incoming calls with no called number in the SETUP message; one of "Accept" or "Reject".
No sub-address	Whether to accept or reject incoming calls with no called sub-address in the SETUP message; one of "Accept" or "Reject".
DLCn	Information about DLC n.
State	The state of the DLC; one of "Initial", "Terminal initiated", "Network initiated" or "OK".
SPID state	The state of the SPID state machine; one of "NULL", "IWAIT1", "IWAIT2", "IWAIT3", "AWAIT1", "AWAIT2", "AWAIT3", "5ESSNOTINIT", "ASPID1", "ASPID2", "ASPID3", "ASPID4", "OP", "5ESSPINIT" or "5ESSMINIT". See Table 4-9 on page 4-19 for a description of these states.
SPID file state	The state of the SPID file state machine; a number in the range 1 to 13. See Table 4-11 on page 4-20 for a description of these states.
Current SPID	The current SPID with which the router is attempting to initialise the DLC.
USID	The User Service Identifier, which identifies the service profile for the interface. This field is only displayed if the State field is set to "OK".
Terminal ID	The Terminal Identifier for the interface. TID values are unique within a given USID. This field is only displayed if the State field is set to "OK".
TSPID	The value of the SPID retry timer.
Common D channel	Parameters concerning non-associated signalling, or common D channel.

Table 4-52: Parameters displayed in the output of the SHOW Q931 command.

Parameter	Meaning
Interface ID	The non-associated signalling, or common D channel, interface identifier.
T301 to T322	The timeout value for the relevant timer.

Figure 4-34: Example output from the SHOW Q931 CALL command.

Inter	Index	State	CallRef	CallRefInit	Timer	ToGo	TOs
0	0	0	0000	USER	—	—	—
0	3	10	0003	USER	—	—	—

Table 4-53: Parameters displayed in the output of the SHOW Q931 CALL command.

Parameter	Meaning
Inter	The ISDN interface.
Index	The call identification number, internal to the router.
State	The state of the call, as per the Q.931 protocol.
CallRef	The call reference as seen by the Q.931 protocol.
CallRefInit	The initiator of the call.
Timer	The timer currently running for this call.
ToGo	The time remaining on the timer.
TOs	The number of timeouts for this timer.

See Also SET Q931

SHOW Q931 SPID

Syntax SHOW Q931 [=interface] SPID

where:

- *interface* is a slotted interface name or number. Interface names are formed by concatenating an interface type and instance (e.g. BRI0 or PRI1). Interface numbers are the decimal index of the slotted interface (0, 1, 2...).

Description This command displays Q.931 SPID information for the specified ISDN interface. The current state of the SPID files for the interface, as well as the state of the SPID state machine and SPID file state machine are displayed (Figure 4-35 on page 4-118, Table 4-54 on page 4-118).

If the auto-SPID procedure is in progress, and the router and network are waiting for user intervention to determine which SPIDs are to be used, then the SPIDs presented by the network will be displayed, along with the bearer capabilities and numbers for those SPIDs. An instructive message which

describes how to enable one or more of the SPIDs is also given (Figure 4-36 on page 4-119, Table 4-55 on page 4-120).

Figure 4-35: Example output from the SHOW Q931 SPID command.

```
Q.931 interface ... BRI0
DLC 1 SPID details
  Number ..... -
  SPID file details
    State ..... 3 (Auto SPID successful)
    Manual SPID .... -
    Generic SPID ... -
    Auto SPID ..... 62155542310101
    Auto BC ..... VDX
  SPID details
    State ..... OP
    Current SPID ... 62155542310101
DLC 2 SPID details
  Number ..... -
  SPID file details
    State ..... 3 (Auto SPID successful)
    Manual SPID .... -
    Generic SPID ... -
    Auto SPID ..... 62155579340101
    Auto BC ..... VD
  SPID details
    State ..... OP
    Current SPID ... 62155579340101

No auto SPID information to display for this interface
```

Table 4-54: Parameters displayed in the output of the SHOW Q931 SPID command.

Parameter	Meaning
Q.931 interface	The name of the Q.931 interface.
DLC <i>n</i> SPID details	Information about DLC (SPID) <i>n</i> .
Number	The directory number for this DLC.
SPID file details	Information about the SPID file for this DLC.
State (SPID file)	The state of the SPID file for this DLC; a number in the range 0 to 13. See Table 4-11 on page 4-20 for a description of these states.
Manual SPID	The manual SPID entered for this DLC.
Generic SPID	The generic SPID obtained from the 10 digit number entered for this DLC.
Auto SPID	The auto SPID selected (either automatically or with manual intervention) for this DLC.
Auto BC	The bearer capabilities associated with the auto SPID for this DLC; one or more of "-" (none), "V" (voice), "D" (data) or "X" (X.25 packet data).
SPID details	Information about the SPID for this DLC.

Table 4-54: Parameters displayed in the output of the SHOW Q931 SPID command.

Parameter	Meaning
State (SPID details)	The state of the SPID initialisation for this DLC; one of "NULL", "IWAIT1", "IWAIT2", "IWAIT3", "AWAIT1", "AWAIT2", "AWAIT3", "5ESSNOTINIT", "ASPID1", "ASPID2", "ASPID3", "ASPID4", "OP", "5ESSPINIT" or "5ESSMINIT". See Table 4-9 on page 4-19 for a description of these states.
Current SPID	The current SPID for this DLC.

Figure 4-36: Example output from the SHOW Q931 SPID command during the auto-SPID procedure.

```

Q.931 interface ... BRI0
DLC 1 SPID details
  Number ..... -
  SPID file details
    State ..... 0 (No SPIDs entered, auto SPID not run or in progress)
    Manual SPID .... -
    Generic SPID ... -
    Auto SPID ..... -
    Auto BC ..... -
  SPID details
    State ..... ASPID3 (manual intervention required)
    Current SPID ... 01010101010101
DLC 2 SPID details
  Number ..... -
  SPID file details
    State ..... 0 (No SPIDs entered, auto SPID not run or in progress)
    Manual SPID .... -
    Generic SPID ... -
    Auto SPID ..... -
    Auto BC ..... -
  SPID details
    State ..... NULL
    Current SPID ... -

```

Auto SPID table for BRI0

Ind	SPID	Bearer	Number	Cause
1	62155542310101	VDX	-	-
2	62155579340101	VD	-	--

Manual intervention is required for one or more of the SPIDs in the table to be selected. Enter the command:

```
ENABLE Q931=0 ASPID=<index>[,<index>]
```

where <index> is the index of the desired auto SPID from the above table. Up to two auto SPIDs may be selected in this fashion.

Table 4-55: Parameters displayed in the output of the SHOW Q931 SPID command during the auto-SPID procedure.

Parameter	Meaning
Q.931 interface	The name of the Q.931 interface.
DLC <i>n</i> SPID details	Information about DLC (SPID) <i>n</i> .
Number	The directory number for this DLC.
SPID file details	Information about the SPID file for this DLC.
State (SPID file)	The state of the SPID file for this DLC; a number in the range 0 to 13. See Table 4-11 on page 4-20 for a description of these states.
Manual SPID	The manual SPID entered for this DLC.
Generic SPID	The generic SPID obtained from the 10 digit number entered for this DLC.
Auto SPID	The auto SPID selected (either automatically or with manual intervention) for this DLC.
Auto BC	The bearer capabilities associated with the auto SPID for this DLC; one or more of "-" (none), "V" (voice), "D" (data) or "X" (X.25 packet data).
SPID details	Information about the SPID for this DLC.
State (SPID details)	The state of the SPID initialisation for this DLC; one of "NULL", "IWAIT1", "IWAIT2", "IWAIT3", "AWAIT1", "AWAIT2", "AWAIT3", "5ESSNOTINIT", "ASPID1", "ASPID2", "ASPID3", "ASPID4", "OP", "5ESSPINIT" or "5ESSMINIT". See Table 4-9 on page 4-19 for a description of these states.
Current SPID	The current SPID for this DLC.
Auto SPID table for <i>interface</i> .	The table of SPID values learned by the auto SPID process.
Ind	The index in the auto SPID table used to select this SPID.
SPID	The SPID value for this auto SPID entry.
Bearer	The bearer capabilities associated with this auto SPID entry; one or more of "-" (none), "V" (voice), "D" (data) or "X" (X.25 packet data).
Number	The directory number associated with this auto SPID entry.
Cause	The cause code associated with this auto SPID entry. A cause of 63 means that the auto SPID is already in use for another device.

Chapter 5

X.25

Introduction	5-2
DTE Mode	5-3
DTE Addresses	5-4
Encapsulations	5-4
Configuring X.25 DTE	5-6
Configure the X.25 DTE Interface	5-6
Configuring Call Parameter Entries	5-7
Configuring Permanent Virtual Circuits	5-7
Configuration Examples	5-7
Command Reference	5-8
ACTIVATE MIOX CIRCUIT	5-9
ADD MIOX CIRCUIT	5-10
ADD X25T CPAR	5-11
CREATE X25T	5-12
DEACTIVATE MIOX CIRCUIT	5-14
DELETE MIOX CIRCUIT	5-15
DELETE X25T CPAR	5-15
DESTROY X25T	5-16
DISABLE MIOX CIRCUIT	5-16
ENABLE MIOX CIRCUIT	5-17
RESET X25T	5-18
SET MIOX	5-18
SET MIOX CIRCUIT	5-19
SET X25T	5-21
SET X25T CPAR	5-22
SHOW MIOX	5-23
SHOW MIOX COUNT	5-24
SHOW MIOX CIRCUIT	5-26
SHOW X25T	5-30
SHOW X25T CPAR	5-34

Introduction

This chapter describes the main features of X.25 Packet Switched Networks, support for X.25 on the router, and how to configure and operate the router to provide, or connect to, an X.25 Packet Switched Network.

CCITT Recommendation X.25 specifies the connection between data terminal equipment (DTE) and data circuit-terminating equipment (DCE) for terminals attached to a public data network. The term X.25 usually refers to Recommendation X.25 plus a number of related CCITT recommendations and other standards produced by organisations such as the Defence Data Network (DDN) and International Standards Organisation (ISO). Since Recommendation X.25 was first produced in 1976, the standard has been adapted for more uses than its original purpose, which was connecting to a public packet switched network.

The general principle of X.25 is that many connections are made over a single physical link. The single link into the data network is maintained at the *data link layer*, while the separate *packet layer* maintains the connections or circuits from one DTE to other DTEs attached to the network. The data link layer is maintained between a DTE and its connected DCE, while packet layer circuits are maintained between pairs of DTEs.

Two main types of circuits exist, *permanent virtual circuits* and *switched virtual circuits*. The permanent circuits are set up by configuration in the DTEs at either end of the circuit and in the X.25 network. When a DTE starts running, it can send data on a permanent circuit straight away. In contrast, switched circuits are not set up when the DTE starts running. Instead a DTE must make a call through the X.25 network to set up the call. A special *call request* packet is sent to the network, containing the DTE address of the DTE being called. DTE addresses are set up by the network administration and uniquely identify DTEs. The DTE being called must accept the call before data transmission can begin. The call may be refused for a number of reasons. Once the call is in progress, data can be exchanged on the switched circuit. Either end of the circuit may terminate the call.

The DTE is the end user equipment that uses the X.25 network. Two common types of DTE are computers and PADs. The term *PAD* means literally *packet assembly/disassembly* but in this context means a terminal server that runs over X.25. A typical PAD has a number of asynchronous ports for supporting terminals and a synchronous port for connecting to the X.25 network. Users of terminals connected to the PAD can call up a remote DTE through the X.25 network.

Computers use X.25 in a number of different ways. Software on the computer can act as a *virtual PAD*, allowing users logging in to the computer to call a remote DTE through the X.25 network. Different software on the computer can use X.25 circuits to set up network connections to other computers for mail and general transfer of data. In all cases the general principle is the same, that of two DTEs making a packet layer connection via the X.25 network.

For more information about the X.25 recommendation and how computers and PADs use X.25, see CCITT recommendations X.25, X.21 and X.121, and ISO 8208.

DTE Mode

The router can be configured to act as a DTE and be attached to an X.25 Packet Switched Network. This allows the routing of information and other protocols through the X.25 network to other routers or DTEs.

The DTE packet layer defines procedures for handling virtual calls over an X.25 network. It provides services to attached modules and specifies the manner in which calls are established, maintained and cleared. When a virtual call is established, a virtual circuit (or call) is set up between the calling and called DTEs. By using a packet-interleaved multiplexing scheme, a single physical circuit can support communications to numerous X.25 DTEs simultaneously. The virtual call is identified at the local DTE-DCE interface by a logical channel number. Each X.25 interface can theoretically support up to 4095 virtual calls.

In order to support multiple virtual circuits over a single data link a unique logical channel number (LCN) is assigned to each virtual call. The significance of LCNs is local between the network components at both ends of the logical data link.

The maximum number of virtual circuits dictates the number of simultaneous sessions that can be supported over a single X.25 data link. Logical channels ranges can be allocated in four categories:

- Permanent virtual circuits.
- Switched virtual circuits for incoming calls only.
- Switched virtual circuits for outgoing calls only.
- Switched virtual circuits for two-way calls only.

A switched virtual circuit (SVC) is required for each virtual call. A virtual call is set up dynamically when one DTE attached to the X.25 network needs to communicate with another DTE. Networks can differentiate between the various types of circuits by the appropriate allocation of logical channel number sequences. When a user places a call either the highest or lowest numbered free channel in the appropriate channel range is selected for the call. If the DTE is connected to a DCE then the highest channel in the particular SVC range is selected. If the interface is in a DTE-to-DTE situation, role negotiation during the restart procedure will determine which DTE will emulate a DCE and it will select channels from the lowest free channel in the particular SVC range.

A permanent virtual circuit (PVC) is essentially a pre-configured permanent logical connection between two end points of a network. It can be likened to a point-to-point leased line. PVCs do not require any connection or disconnection procedures and data can be sent on PVCs as soon as the X.25 link is active.

The router supports both permanent and switched virtual circuits. Having set up an X.25 DTE interface, the router can be linked to the X.25 network, and it may then make X.25 calls to other routers or DTE equipment.

When the X.25 DTE interface is operational, IP can be initialised to run over the X.25 network (see *Chapter 6, Internet Protocol (IP)* for further information about configuring IP to run over X.25).

Features of the router's implementation of an X.25 DTE include:

- Packet layer modulus of 8 or 128.

- Packet layer window size of 1 to 7 for modulo 8 and 1 to 127 for modulo 128.
- Variable packet size supporting data packets from 128 to 1024 bytes.
- Role determination for DTE-to-DTE connections.
- There is currently no support for call facilities.
- Ability to transport IP over X.25.

DTE Addresses

The syntax of DTE addresses is covered in Recommendation X.25. A related standard, Recommendation X.121, specifies a numbering plan for public data networks that follows the format outlined in X.25. To access a DTE on a public data network, an address conforming to X.121 must be used. However, if a private X.25 network is being run over a network of routers, any DTE addresses may be used.

A DTE address consists of 1 to 15 decimal digits. On the router any X.25 interface can be set up to respond to any DTE address. A single interface can be set up to respond to a number of different DTE addresses. Wildcards may also be used in the DTE address to further expand the addresses to which an interface will respond. This facility has been provided for two reasons:

- Some PADs respond to a range of DTE addresses, for example in accessing particular ports on the PAD.
- The router may be attached to an X.25 gateway, which may, for example, be providing access to a public X.25 network.

An example of DTE wildcard addressing is an interface set up to respond to DTE addresses 12345, 12346 and 2345X. The last address contains the wildcard character 'X'. The interface with these addresses will respond to any calls to any of these addresses, with the wildcard address mapping all addresses from 23450 to 23459.

Encapsulations

The MIOX (*Multiprotocol Interconnect on X.25*) encapsulation scheme defined in RFC 1356, "*Multiprotocol Interconnect on X.25 and ISDN in the Packet Mode*", specifies the encapsulation of IP and other network layer protocols over X.25 networks in accordance with ISO/IEC and CCITT standards.

Virtual circuits are opened on demand when datagrams arrive at the network interface for transmission, and are closed after a specified period of activity or when the interface runs out of virtual circuits. When an attempt is made to open a virtual circuit, the first octet of the Call User Data field of the Call Request packet is a Network Layer Protocol Identifier (NLPID) used to specify the protocol encapsulation to be used on the virtual circuit (Table 5-1 on page 5-5).

Table 5-1: NLPID values for protocol encapsulation over X.25 circuits.

NLPID (hexadecimal)	Protocol Encapsulation
CC	Internet Protocol (IP)
81	CLNP
82	ES-IS
80	SNAP (Subnetwork Address Protocol)
00	Null encapsulation

If the SNAP encapsulation is used, the five octets following the NLPID in the Call User Data field contain the SNAP header for the network layer protocol. Only a single protocol may be carried over a virtual circuit using the IP, CLNP, ES-IS or SNAP encodings. Multiple virtual circuits must be established to transport multiple protocols over the same X.25 connection. Once the circuit has been established, the data field of X.25 data packets contains only Protocol Data Units (PDUs) for the specified network layer protocol.

The Null encapsulation may be used to multiplex multiple network layer protocols over a single virtual circuit. The NLPID in the Call User Data field contains the null value, and the first octet of the data field of each X.25 data packet contains the NLPID followed by the PDU for the specified network layer protocol.



Only the IP and NULL encapsulations are supported.



The same encapsulation must be specified on the routers at each end of the circuit. If different encapsulations are used the routers will not be able to communicate with one another over the MIOX circuit.

A MIOX circuit is created with the command:

```
ADD MIOX=x25t-interface CIRCUIT=circuit-name
    ENCAP=encapsulation
```

and deleted with the command:

```
DELETE MIOX=x25t-interface CIRCUIT=circuit-name
```

A MIOX circuit can be explicitly activated and deactivated to test the configuration of the circuit without generating protocol traffic, using the commands:

```
ACTIVATE MIOX=x25t-interface CIRCUIT=circuit-name
DEACTIVATE MIOX=x25t-interface CIRCUIT=circuit-name
```

A MIOX circuit may be temporarily disabled or enabled, without losing the configuration, using the commands:

```
DISABLE MIOX=x25t-interface CIRCUIT=circuit-name
ENABLE MIOX=x25t-interface CIRCUIT=circuit-name
```

Parameters that affect the operation of the MIOX module and individual MIOX circuits may be changed after the MIOX circuits have been created, using the commands:

```
SET MIOX=x25t-interface
SET MIOX=x25t-interface CIRCUIT=circuit-name
```

The status of the MIOX module and individual MIOX circuits can be displayed with the commands:

```
SHOW MIOX=[x25t-interface]
SHOW MIOX=[x25t-interface] COUNT
SHOW MIOX=[x25t-interface] CIRCUIT=[circuit-name]
```

The IP module can be configured to use a MIOX circuit with the command:

```
ADD IP ARP=ipadd INTERFACE=interface CIRCUIT=circuit-name
```

Configuring X.25 DTE

The following steps are required to set up X.25 DTE on the router.

- Configure an X.25 DTE interface to run over a LAPD interface.
- Configure Call Parameter entries.
- Configure Permanent Virtual Circuits.

Note that if the default parameters are satisfactory then it may not be necessary to configure the parameters for an X.25 DTE interface.

Configure the X.25 DTE Interface

To configure an interface to run X.25 DTE, use the command:

```
CREATE X25T=interface OVER=LAPDn
```

Note that the X.25 DTE interface number is not directly related to the LAPD interface number.

The parameters that define the operation of an X.25 DTE interface can either be set when the X.25 DTE interface is created, or by using the command:

```
SET X25T
```

An X.25 DTE interface, when first created has operational parameters set to default values that may need to be altered for communication over the interface to begin.

It is essential that certain operational parameters are identical between the DTE interface and the X.25 DCE network or peer DTE to which it is connected.

The packet modulus must be identical at both ends of the link and the logical channel number ranges must be consistent to provide correct identification of incoming calls.

The X.25 DTE interface must be configured with a DTE address before any data traffic can be transmitted over the X.25 DTE interface. This address is placed in the calling field of outgoing call request packets and provides peer DTEs with an indication of where the call originated.

The X.25 DTE interface timers must be set up correctly so that the various timers allow enough time to correctly deal with packet retransmissions and other timing constraints.

When the operational parameters are altered for a X.25 DTE interface the changes do not take place immediately, and are not reflected in the SHOW X25T command on page 5-30 until the X.25 DTE interface is reset, using the command:

```
RESET X25T
```

The RESET command updates all changes to operational parameters and initiates the X.25 DTE link restart procedure.

Configuring Call Parameter Entries

Call parameter entries are used to specify parameters to use for individual calls. Call parameter entries contain information about the data and window sizes to be used by a X.25 call. To create a call parameter entry use the following command:

```
ADD X25T CPAR
```

The following command creates a call parameter entry and any calls that reference this entry will use the specified operational parameters for that call:

```
ADD X25T CPAR=1 MAXDATA=128 WINDOW=4
```

A maximum of 8 call parameter entries can be configured.

Configuring Permanent Virtual Circuits

When PVCs (*Permanent Virtual Circuits*) are used these must be configured as a MIOX circuit, using the command:

```
ADD MIOX=x25t-interface CIRCUIT=circuit-name PVC=0..4095
```

PVC channels are supplied by the X.25 network and are configured to provide a direct line to a peer DTE. To configure PVCs the DTE interface logical channel range parameters must be set up correctly, using the command:

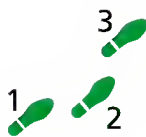
```
CREATE X25T=x25-interface OVER={LABPBn|LAPDn} NPVC=0..4095
```

Configuration Examples

The following example illustrates the steps required to configure an X.25 network over the ISDN D channel.

X.25 over the D Channel allows ISDN customers to make and receive packet switched calls using the ISDN D Channel as an access point. This provides approximately a 4800 bps switched connection that the router can use to transport IP traffic over X.25.

This example illustrates how to configure an X.25 DTE interface over the ISDN D channel, and then configure IP to use the X.25 interface.



To configure X.25 over the ISDN D channel:

1. Create an X.25 DTE interface over the ISDN D channel.

Create X.25 DTE interface 0 over the D channel of ISDN interface 0, which supports X.25 packet network access:

```
CREATE X25T=0 OVER=LAPD0
```

2. Configure the X.25 DTE interface.

Assign logical channel numbers 1 and 2 to two-way channels to correspond with the channel ranges supplied by the network:

```
SET X25T=0 LTC=1 HTC=2 DTE=123456
```

Set the packet window and packet length parameters if the network values for these parameters are different from the default router configuration. Default values for the router are a window size of 2 and a packet length of 128 octets. If the network supplied values are different a call parameter must be created to specify the required values. For example, to create a call parameter with an index number of 1, a packet length of 256 octets and a window size of 4, use the command:

```
ADD X25T CPAR=1 MAXDATA=256 WINDOW=4
```

Once the call parameter is created the interface must be configured to use the call parameters for the default values and reset for the changes to take effect. To configure DTE interface 0 to use call parameter set 1, use the command:

```
SET X25T=0 DEFCPAR=1  
RESET X25T=0
```

3. Configure the IP routing module.

Create an IP interface for X.25 DTE interface 0, with an IP address of 192.168.35.101:

```
ADD IP INT=X25T0 IP=192.168.35.101 MASK=255.255.255.240
```

To enable IP traffic to be transmitted over the X.25 DTE interface, ARP entries must be created for other routers that are accessible via the X.25 DTE interface, using the command:

```
ADD MIOX=0 CIRC=TEST DTE=54321  
ADD IP ARP=192.168.35.102 INT=X25T0 CIRC=TEST
```

The IP parameter specifies the IP address of the remote router's X.25 DTE interface and the DTE parameter specifies the ISDN DTE address of the router's ISDN connection. ARP entries must be created for every remote router that the router will make switched calls to via X.25 over the D Channel. The remote routers must also have ARP entries specifying this router's X.25 DTE IP address and DTE number.

Command Reference

This section describes the commands available on the router to set up and configure the router as an X.25 DTE.

See *"Conventions"* on page xxxv of *Preface* in the front of this manual for details of the conventions used to describe command syntax. See *Appendix A, Messages* for a complete list of messages and their meanings.

ACTIVATE MIOX CIRCUIT

Syntax `ACTIVATE MIOX=x25t-interface CIRCUIT=circuit-name`
 `[USER=IP]`

where:

- *x25t-interface* is the index number of the X.25 DTE logical interface, in the range 0 to 7.
- *circuit-name* is an alphanumeric string, 1 to 15 characters in length.

Description This command activates a MIOX circuit, forcing an X.25 call to be set up to the remote router without the need to generate protocol traffic. This command can be used to test that the MIOX circuit has been correctly configured at both ends of the X.25 DTE link. If the circuit is already open or is attempting to open the command is ignored.



This command is only relevant for circuits configured over SVCs.

The MIOX parameter specifies the X.25 DTE interface over which the MIOX circuit has been defined. The X.25 DTE interface must already exist.

The CIRCUIT parameter specifies the name of the MIOX circuit to activate. The circuit name must already exist for the interface and must specify an SVC circuit. The circuit must be enabled and have a user module attached to it.

The USER parameter specifies the user module and must be used if the circuit encapsulation supports multiple circuits. MIOX will activate an X.25 call for the specified user module. The user module must be attached to the circuit. The USER parameter is only required for circuits with multiple encapsulations.

Examples To activate the MIOX circuit “HeadOffice” on X.25 DTE interface 1, use the command:

```
ACTIVATE MIOX=1 CIRCUIT=HeadOffice
```

See Also `ADD MIOX CIRCUIT`
 `DEACTIVATE MIOX CIRCUIT`
 `DELETE MIOX CIRCUIT`
 `DISABLE MIOX CIRCUIT`
 `ENABLE MIOX CIRCUIT`
 `SET MIOX CIRCUIT`
 `SHOW MIOX CIRCUIT`

ADD MIOX CIRCUIT

Syntax `ADD MIOX=x25t-interface CIRCUIT=circuit-name
{DTEADDRESS=dteaddress|PVC=1..4095} [CPAR=0..8]
[ENCAP={IP|NULL|MULTIPLE}] [COMMENT=comment] [COMP={ON|
OFF}] [TCPCOMP={ON|OFF}]`

where:

- *x25t-interface* is the index number of the X.25 DTE logical interface, in the range 0 to 7.
- *circuit-name* is an alphanumeric string, 1 to 15 characters in length.
- *dteaddress* is a valid DTE address, 1 to 15 characters in length.
- *comment* is an alphanumeric string, 1 to 40 characters in length.

Description This command adds a MIOX circuit over the X.25 DTE interface. Adding a MIOX circuit to an interface allows the router to transport protocols over the X.25 packet network to a remote router. A MIOX circuit, with appropriate local configuration settings, must be added to the routers at both ends of the X.25 DTE link for communication to take place.

The MIOX parameter specifies the X.25 DTE interface over which the MIOX circuit has been defined. The X.25 DTE interface must already exist.

The CIRCUIT parameter specifies the circuit name of the MIOX circuit to add, and must identify a unique circuit. The MIOX circuit name must not exist for the interface.

The DTEADDRESS parameter specifies the DTE address of the remote router, if it is to be accessed via an SVC (*switched virtual circuit*). The DTE address must not be in use by any other MIOX circuits. Any calls made to the remote router will use the specified DTE address to make the call.

The PVC parameter creates a PVC channel to the remote router to be used by this MIOX circuit. The PVC channel must be valid for the X.25 DTE interface and must not be in use by another circuit or module.

Either DTEADDRESS or PVC must be specified, but not both. A circuit can be supported over a PVC or an SVC, but not both simultaneously.

The CPAR parameter specifies the index for the call parameters to use for the MIOX circuit. If a value is not specified, the default is the default call parameters assigned for the X.25 DTE interface in the last CREATE X25T or SET X25T command executed. If a call parameter index is specified and is not 0, the call parameter index must exist. The default is 0.

The ENCAP parameter specifies the encapsulation to use. If MULTIPLE is specified, one X.25 channel is used for each higher layer protocol (i.e. IP). This may result in several circuits being opened to a remote router, each carrying a different protocol. If NULL is specified, the Null encapsulation is used to multiplex more than one higher layer protocol over a single X.25 circuit to the remote router associated with the circuit. If IP is specified, the circuit will support only the IP encapsulation. Calls received for other protocols will be rejected. If the MIOX circuit is connected over a PVC then only IP or NULL may be specified, since the MULTIPLE option requires several X.25 channels and cannot therefore run over a PVC. The default is MULTIPLE for SVCs and NULL for PVCs.



The same encapsulation must be specified on the routers at each end of the circuit. If different encapsulations are used the routers will not be able to communicate with one another over the MIOX circuit.

The COMMENT parameter specifies a string that provides a textual description of the MIOX circuit and is displayed in the output of the SHOW MIOX CIRCUIT command on page 5-26.

The COMP parameter enables or disables the use of packet compression for the MIOX circuit. When packet compression is used the remote router must be a compatible router that also supports the use of compression. The default is OFF.

The TCPCOMP parameter enables or disables the use of TCP header compression. This is only relevant if the circuit encapsulation supports IP. Van Jacobson's compression is used. Compression provides the most advantage on slower link speeds (up to 48 kbps). At speeds of 64 kbps and higher, compression will actually reduce efficiency and so should be disabled. For successful TCP header compression the remote router must also have TCP compression enabled for its corresponding circuit. The default is OFF.

Examples To create a MIOX circuit called "RemoteOffice" on X.25 DTE interface 0 using PVC 1 and the NULL encapsulation, use the command:

```
ADD MIOX=1 CIRCUIT=RemoteOffice PVC=1 ENCAP=NULL
```

See Also ACTIVATE MIOX CIRCUIT
DEACTIVATE MIOX CIRCUIT
DELETE MIOX CIRCUIT
DISABLE MIOX CIRCUIT
ENABLE MIOX CIRCUIT
SET MIOX CIRCUIT
SHOW MIOX CIRCUIT

ADD X25T CPAR

Syntax ADD X25T CPAR=*call-index* [COPY=*call-index*] [MAXDATA={128|256|512|1024}] [NUI=*nui*] [RMAXDATA={128|256|512|1024}] [RWINDOW=1..127] [TMAXDATA={128|256|512|1024}] [TWINDOW=1..127] [USERDATA=*hex-string*] [WINDOW=1..127]

where:

- *call-index* is the index of the new call parameters or the call parameters to copy.
- *nui* is a character string, 1 to 16 characters in length.
- *hex-string* is a string of 2 to 20 hexadecimal digits.

Description This command creates a set of call parameters for X.25. The call parameter must not already exist. The COPY parameter specifies a set of existing call parameters to copy as the default values for this call parameter set. The COPY parameter, if present, must immediately follow the CPAR parameter. Any parameters after COPY modify the copy of the call parameter set.

The NUI parameter specifies the Network User Identification (NUI) to be used by this call.

The TMAXDATA and RMAXDATA parameters specify the maximum packet sizes for transmission and reception, respectively. The MAXDATA parameter is a shorthand form to specify the same maximum packet size for both transmission and reception. In all cases, the default is 0, which means use the interface default for a call (128).

The TWINDOW and RWINDOW parameters specify the window for transmission and reception, respectively. The WINDOW parameter is a shorthand form to specify the same window for both transmission and reception. In all cases, the default is 0, which means use the interface default for a call (2).

The USERDATA parameter specifies the contents of the user data field for the call request for this call. This is used to specify the protocol that this circuit will carry.

Examples To create call parameter set 1 for X.25 DTE interfaces that uses a maximum packet size of 256 bytes and a window size of 4 for both transmission and reception, use the command:

```
ADD X25T CPAR=1 MAXDATA=256 WINDOW=4
```

See Also CREATE X25T
DELETE X25T CPAR
SET X25T CPAR
SHOW X25T CPAR

CREATE X25T

Syntax `CREATE X25T=x25-interface OVER=LAPDn [MAXACTIVE=0..4095]
[MODULUS={8|128}] [T20=1..360] [T21=1..360] [T22=1..360]
[T23=1..360] [T24={1..360|OFF}] [T27={1..360|OFF}]
[MINRECALL=1..360] [R20=0..65535] [R22=0..65535]
[R23=0..65535] [R27=0..65535] [NPVC=0..4095]
[DEFCPAR=0..8] [DTEADDRESS=dteaddress] [LIC=0..4095]
[HIC=0..4095] [LTC=0..4095] [HTC=0..4095] [LOC=0..4095]
[HOC=0..4095] [ROLE={DYNAMIC|DCE|DTE}]`

where:

- *x25-interface* is the number of the X.25 DTE logical interface, in the range 0 to 7.
- *n* is the number of the LAPD interface over which the X.25 interface will run.
- *dteaddress* is a valid DTE address, 1 to 15 characters in length.

Description This command creates an instance of the X.25 DTE module, that is, an X.25 interface. The X.25 interface must not already exist.

The OVER parameter specifies the layer 2 entity (LAPD) to be used by the X.25 interface. The layer 2 entity must already exist and must not be in use by another X.25 DTE interface.

The MAXACTIVE parameter sets the maximum number of circuits this X.25 DTE can support, including PVCs. It must not be less than the actual number of PVCs. The default value is 4095. The X.25 DTE interface must be reset with the RESET X25T command on page 5-18 before this command takes effect, since there may be more than the maximum number of circuits active already.

The MODULUS parameter sets the modulus of packet sequence numbers for this interface. The default value is 8. The X.25 DTE interface must be reset with the RESET X25T command on page 5-18 before this command takes effect, since this command affects all circuits on the interface.

The T20 parameter sets the value in seconds of the restart timer. The default value is 180. The T21 parameter sets the value in seconds of the call timer. The default value is 200. The T22 parameter sets the value in seconds of the reset timer. The default value is 180. The T23 parameter sets the value in seconds of the clear timer. The default value is 180. The T24 parameter sets the value in seconds of the optional window timer. The default value is 60. The timer can be turned off by specifying the value OFF. The T27 parameter sets the value in seconds of the reject response timer. The default value is 60. The timer can be turned off by specifying the value OFF. The MINRECALL parameter sets the value in seconds of the time to wait before retrying a particular call. The default value is 60.

The R20 parameter sets the value of the restart retransmission counter. The default value is 1. The R22 parameter sets the value of the reset request retransmission counter. The default value is 1. The R23 parameter sets the value of the clear request retransmission counter. The default value is 1. The R27 parameter sets the value of the reject retransmission counter. The default value is 0.

The NPVC parameter sets the number of channels reserved as PVCs. The default is 0.

The DEFPCPAR parameter specifies the index of the default call definition to use for this interface. The default value is 0, which means no default call. The call definition (if not 0) must already exist.

The DTEADDRESS parameter specifies the DTE address for the interface.

The LIC parameter specifies the lowest incoming channel number. The default is 0. The HIC parameter specifies the highest incoming channel number. A value of 0 means no incoming channels. The default is 0. The LTC parameter specifies the lowest two-way channel number. The default is 1. The HTC parameter specifies the highest two-way channel number. A value of 0 means no two-way channels. The default is 4095. The LOC parameter specifies the lowest outgoing channel number. The default is 0. The HOC parameter specifies the highest outgoing channel number. A value of 0 means no outgoing channels. The default is 0. The values of LIC, HIC, LTC, HTC, LOC, LTC, NPVC and MAXACTIVE must be consistent with one another.

The ROLE parameter specifies the role this DTE plays in a DTE-DTE environment. The default is DYNAMIC, which means during the restart procedure the two DTEs will negotiate which DTE will act as DCE.

Examples To create X.25 DTE interface 0 over LAPD interface 0 with PVCs on channels 1 to 5, two-way SVCs on channels 6 to 15 and default call parameter set 1, use the command:

```
CREATE X25T=0 OVER=LAPD0 NPVC=5 LTC=6 HTC=15 DEFPCPAR=1
```

See Also ADD X25T CPAR
DESTROY X25T
SET X25T
SET X25T CPAR
RESET X25T
SHOW X25T
SHOW X25T CPAR

DEACTIVATE MIOX CIRCUIT

Syntax DEACTIVATE MIOX=*x25t-interface* CIRCUIT=*circuit-name*
[USER=IP]

where:

- *x25t-interface* is the index number of the X.25 DTE logical interface, in the range 0 to 7.
- *circuit-name* is an alphanumeric string, 1 to 15 characters in length.

Description This command deactivates a MIOX circuit. When a circuit has an active X.25 call this command can be used to closed the call to the remote router. If the circuit is already closed the command is ignored.

The MIOX parameter specifies the X.25 DTE interface over which the MIOX circuit has been defined. The X.25 DTE interface must already exist.

The CIRCUIT parameter specifies the name of the MIOX circuit to deactivate. The circuit name must already exist for the interface and must specify an SVC circuit. The circuit must be enabled and have a user module attached to it.

The USER parameter specifies the user module and must be used if the circuit encapsulation supports multiple circuits. MIOX will deactivate the X.25 call for the specified user module. The user module must be attached to the circuit. The USER parameter is only required for circuits with multiple encapsulations.

Examples To deactivate the MIOX circuit "HeadOffice" on X.25 DTE interface 1, use the command:

```
DEACTIVATE MIOX=1 CIRCUIT=HeadOffice
```

See Also ACTIVATE MIOX CIRCUIT
ADD MIOX CIRCUIT
DELETE MIOX CIRCUIT
DISABLE MIOX CIRCUIT
ENABLE MIOX CIRCUIT
SET MIOX CIRCUIT
SHOW MIOX CIRCUIT

DELETE MIOX CIRCUIT

Syntax `DELETE MIOX=x25t-interface CIRCUIT=circuit-name`

where:

- *x25t-interface* is the index number of the X.25 DTE logical interface, in the range 0 to 7.
- *circuit-name* is an alphanumeric string, 1 to 15 characters in length.

Description This command deletes a MIOX circuit. Deleting a MIOX circuit from an interface will destroy the connection to the remote router. When a MIOX circuit is deleted the remote router's corresponding MIOX circuit should also be deleted. A MIOX circuit can not be deleted if there are any protocols currently using the circuit. Any configured protocols must be detached before deleting the circuit.

The MIOX parameter specifies the X.25 DTE interface over which the MIOX circuit has been defined. The X.25 DTE interface must already exist.

The CIRCUIT parameter specifies the name of the MIOX circuit to delete. The circuit name must already exist for the interface. The circuit must not have any user modules attached to it.

Examples To delete the MIOX circuit "HeadOffice" on X.25 DTE interface 1, use the command:

```
DELETE MIOX=1 CIRCUIT=HeadOffice
```

See Also `ACTIVATE MIOX CIRCUIT`
`ADD MIOX CIRCUIT`
`DEACTIVATE MIOX CIRCUIT`
`DISABLE MIOX CIRCUIT`
`ENABLE MIOX CIRCUIT`
`SET MIOX CIRCUIT`
`SHOW MIOX CIRCUIT`

DELETE X25T CPAR

Syntax `DELETE X25T CPAR=call-index`

where:

- *call-index* is the index of the call parameters to delete.

Description This command deletes a set of call parameters for X.25. The call parameters must already exist, and must not be the default for an X.25 DTE interface or PVC, or be in use by an active call.

Examples To delete X.25 DTE call parameter set 1, use the command:

```
DELETE X25T CPAR=1
```

See Also ADD X25T CPAR
SET X25T CPAR
SHOW X25T CPAR

DESTROY X25T

Syntax DESTROY X25T=*x25-interface*

where:

- *x25-interface* is the number of the X.25 DTE logical interface, in the range 0 to 7.

Description This command destroys an X.25 DTE interface. The X.25 DTE interface must already exist. The interface will not be destroyed if higher layer entities are attached to the X.25 DTE interface.

Examples To destroy X.25 DTE interface 1, use the command:

```
DESTROY X25T=1
```

See Also CREATE X25T
SET X25T
RESET X25T
SHOW X25T

DISABLE MIOX CIRCUIT

Syntax DISABLE MIOX=*x25t-interface* CIRCUIT=*circuit-name*

where:

- *x25t-interface* is the index number of the X.25 DTE logical interface, in the range 0 to 7.
- *circuit-name* is an alphanumeric string, 1 to 15 characters in length.

Description This command disables a previously enabled MIOX circuit. Once a circuit is disabled X.25 calls will not be attempted or accepted and data will not be exchanged with the remote router. If the call is active and is an SVC the call will be closed.

The MIOX parameter specifies the X.25 DTE interface over which the MIOX circuit has been defined. The X.25 DTE interface must already exist.

The CIRCUIT parameter specifies the name of the MIOX circuit to disable. The circuit name must already exist for the interface.

Examples To temporarily disable MIOX circuit "HeadOffice" on X.25 DTE interface 1, use the command:

```
DISABLE MIOX=1 CIRCUIT=HeadOffice
```

See Also ACTIVATE MIOX CIRCUIT
 ADD MIOX CIRCUIT
 DEACTIVATE MIOX CIRCUIT
 DELETE MIOX CIRCUIT
 ENABLE MIOX CIRCUIT
 SET MIOX CIRCUIT
 SHOW MIOX CIRCUIT

ENABLE MIOX CIRCUIT

Syntax `ENABLE MIOX=x25t-interface CIRCUIT=circuit-name`

where:

- *x25t-interface* is the index number of the X.25 DTE logical interface, in the range 0 to 7.
- *circuit-name* is an alphanumeric string, 1 to 15 characters in length.

Description This command enables a previously disabled MIOX circuit. When MIOX circuits are added to the X.25 DTE interface they are by default enabled. Once a circuit is enabled X.25 calls will be attempted or accepted and data will be exchanged with the remote router.

The MIOX parameter specifies the X.25 DTE interface over which the MIOX circuit has been defined. The X.25 DTE interface must already exist.

The CIRCUIT parameter specifies the name of the MIOX circuit to enable. The circuit name must already exist for the interface.

Examples To enable MIOX circuit "HeadOffice" on X.25 DTE interface 1, use the command:

```
ENABLE MIOX=1 CIRCUIT=HeadOffice
```

See Also ACTIVATE MIOX CIRCUIT
 ADD MIOX CIRCUIT
 DEACTIVATE MIOX CIRCUIT
 DELETE MIOX CIRCUIT
 DISABLE MIOX CIRCUIT
 SET MIOX CIRCUIT
 SHOW MIOX CIRCUIT

RESET X25T

Syntax RESET X25T=*x25-interface*

where:

- *x25-interface* is the number of the X.25 DTE logical interface, in the range 0 to 7.

Description This command resets an X.25 DTE instance. The X.25 DTE interface must already exist. Two copies of an X.25 interface's parameters exist, the administrative copy and the operational copy. The administrative copy is altered by the ADD, CREATE, DELETE and SET commands. The operational copy is the copy used for the actual operation of the interface. The RESET command copies the administrative parameters to the operational parameters, clears all switched circuits, resets all PVCs, and restarts the interface. The RESET command must be used after changes have been made to X.25 parameters for those changes to take effect.

Examples To reset X.25 DTE interface 1 to its configured settings, use the command:

```
RESET X25T=1
```

See Also CREATE X25T
DESTROY X25T
SET X25T
SHOW X25T

SET MIOX

Syntax SET MIOX=*x25t-interface* [MINOPEN=10..360]
[INACTIVE=10..360] [HOLDDOWN=10..360]
[COLLISION=10..360] [FAILURE=10..360]

where:

- *x25t-interface* is the index number of the X.25 DTE logical interface, in the range 0 to 7.

Description This command sets the operational parameters for the MIOX entity associated with an X.25 DTE interface. The X.25 DTE interface must already exist. All parameters are timer values that are used to determine the operation of switched virtual calls over the X.25 DTE interface. Alterations to the MIOX parameters take effect immediately. All MIOX timer parameters are specified in seconds.

The MINOPEN parameter sets the minimum time, in seconds, to keep a call open, once the call has been successfully established. After the specified period of time the MIOX layer will close the call provided the link has been idle for a period of time specified by the INACTIVE parameter. The default value is 60 seconds.

The INACTIVE parameter specifies the length of time, in seconds, a call must remain inactive before it is closed. An call is considered inactive when there has been no data transmitted or received over that call. The call will not be closed if

the MINOPEN parameter would be compromised by doing so. The default value is 60 seconds.

The HOLDDOWN parameter sets the minimum length of time, in seconds, to wait after a call has failed before retrying the call. The default value is 60 seconds.

The COLLISION parameter specifies the length of time, in seconds, to wait before retrying a call after a call failed due to a lack of available X.25 channels. The default value is 60 seconds.

The FAILURE parameter sets the length of time, in seconds, to wait before considering an attempted outgoing call to have failed. The call will be retried after a period of time specified by the HOLDDOWN parameter. The default value is 60 seconds.

Examples To configure the MIOX circuits on X.25 DTE interface 1 to disconnect after only 30 seconds of inactivity, use the command:

```
SET MIOX=1 INACTIVE=30
```

See Also SHOW MIOX

SET MIOX CIRCUIT

Syntax `SET MIOX=x25t-interface CIRCUIT=circuit-name
[{DTEADDRESS=dteaddress | PVC=1..4095}] [CPAR=0..8]
[ENCAP={IP | NULL | MULTIPLE}] [COMMENT=comment] [COMP={ON |
OFF}] [TCPCOMP={ON | OFF}]`

where:

- *x25t-interface* is the index number of the X.25 DTE logical interface, in the range 0 to 7.
- *circuit-name* is an alphanumeric string, 1 to 15 characters in length.
- *dteaddress* is a valid DTE address, 1 to 15 characters in length.
- *comment* is an alphanumeric string, 1 to 40 characters in length.

Description This command sets the operational parameters of a MIOX circuit. The circuit must already exist. Changes to MIOX circuit parameters take effect immediately. Equivalent changes must be made to the MIOX circuits on the routers at both ends of the X.25 DTE link for communication to take place.

The MIOX parameter specifies the X.25 DTE interface over which the MIOX circuit has been defined. The X.25 DTE interface must already exist.

The CIRCUIT parameter specifies the circuit name of the MIOX circuit, and must identify a unique circuit. The MIOX circuit name must already exist for the interface.

The DTEADDRESS parameter specifies a DTE address for the remote router, if it is to be accessed via an SVC (*switched virtual circuit*). The DTE address must not be in use by any other MIOX circuits. Changing the DTE address will cause any currently active calls to the old DTE address to be closed. Any calls made

to the remote router after the successful execution of this command will use the DTE address to make the call. If the circuit was previously active over an X.25 PVC then the circuit will detach from the PVC.

The PVC parameter creates a PVC channel to the remote router to be used by this MIOX circuit. The PVC channel number must be valid for the X.25 DTE interface and must not be in use by another circuit or module. Changing the PVC channel will close any current calls to the remote router and detach the circuit from the old PVC.

Either DTEADDRESS or PVC must be specified, but not both. A circuit can be supported over a PVC or an SVC, but not both simultaneously.

The CPAR parameter specifies the index for the call parameters to use for the MIOX circuit. If a value is not specified, the default is the default call parameters assigned for the X.25 DTE interface in the last CREATE X25T command or SET X25T command executed. If a call parameter index is specified and is not 0, the call parameter index must exist.

The ENCAP parameter specifies the encapsulation to use. If MULTIPLE is specified, one X.25 channel is used for each higher layer protocol (i.e. IP). This may result in several circuits being opened to a remote router, each carrying a different protocol. If NULL is specified, the Null encapsulation is used to multiplex more than one higher layer protocol over a single X.25 circuit to the remote router associated with the circuit. If IP is specified, the circuit will support only the IP encapsulation. Calls received for other protocols will be rejected. If the MIOX circuit is connected over a PVC then only IP or NULL may be specified, since the MULTIPLE option requires several X.25 channels and cannot therefore run over a PVC.



The same encapsulation must be specified on the routers at each end of the circuit. If different encapsulations are used the routers will not be able to communicate with one another over the MIOX circuit.

The COMMENT parameter specifies a string that provides a textual description of the MIOX circuit and is displayed in the output of the SHOW MIOX CIRCUIT command on page 5-26.

The COMP parameter enables or disables the use of packet compression for the MIOX circuit. When packet compression is used the remote router must be a compatible router that also supports the use of compression.

The TCPCOMP parameter enables or disables the use of TCP header compression. This is only relevant if the circuit encapsulation supports IP. Van Jacobson's compression is used. Compression provides the most advantage on slower link speeds (up to 48 kbps). At speeds of 64 kbps and higher, compression will actually reduce efficiency and so should be disabled. For successful TCP header compression the remote router must also have TCP compression enabled for its corresponding circuit.

Examples To set the encapsulation to IP and enable TCP header compression for MIOX circuit "HeadOffice" on X.25 DTE interface 1, use the command:

```
SET MIOX=1 CIRCUIT=HeadOffice ENCAP=IP TCPCOMP=ON
```

See Also ACTIVATE MIOX CIRCUIT
 ADD MIOX CIRCUIT
 DEACTIVATE MIOX CIRCUIT
 DELETE MIOX CIRCUIT
 DISABLE MIOX CIRCUIT
 ENABLE MIOX CIRCUIT
 SHOW MIOX CIRCUIT

SET X25T

Syntax SET X25T=*x25-interface* [MAXACTIVE=0..4095] [MODULUS={8|128}] [T20=1..360] [T21=1..360] [T22=1..360] [T23=1..360] [T24={1..360|OFF}] [T27={1..360|OFF}] [MINRECALL=1..360] [R20=0..65535] [R22=0..65535] [R23=0..65535] [R27=0..65535] [NPVC=0..4095] [DEFPCPAR=0..8] [DTEADDRESS=*dteaddress*] [LIC=0..4095] [HIC=0..4095] [LTC=0..4095] [HTC=0..4095] [LOC=0..4095] [HOC=0..4095] [ROLE={DYNAMIC|DCE|DTE}]

where:

- *x25-interface* is the number of the X.25 DTE logical interface, in the range 0 to 7.
- *dteaddress* is a valid DTE address, 1 to 15 characters in length.

Description This command is used to change the operational parameters of an X.25 DTE interface. The X.25 interface must already exist.

The MAXACTIVE parameter sets the maximum number of circuits this X.25 DTE can support, including PVCs. It must not be less than the actual number of PVCs. The default value is 4095. The X.25 DTE interface must be reset with the RESET X25T command on page 5-18 before this command takes effect, since there may be more than the maximum number of circuits active already.

The MODULUS parameter sets the modulus of packet sequence numbers for this interface. The default value is 8. The X.25 DTE interface must be reset with the RESET X25T command on page 5-18 before this command takes effect, since this command affects all circuits on the interface.

The T20 parameter sets the value in seconds of the restart timer. The default value is 180. The T21 parameter sets the value in seconds of the call timer. The default value is 200. The T22 parameter sets the value in seconds of the reset timer. The default value is 180. The T23 parameter sets the value in seconds of the clear timer. The default value is 180. The T24 parameter sets the value in seconds of the optional window timer. The default value is 60. The timer can be turned off by specifying the value OFF. The T27 parameter sets the value in seconds of the reject response timer. The default value is 60. The timer can be turned off by specifying the value OFF. The MINRECALL parameter sets the value in seconds of the time to wait before retrying a particular call. The default value is 60.

The R20 parameter sets the value of the restart retransmission counter. The default value is 1. The R22 parameter sets the value of the reset request retransmission counter. The default value is 1. The R23 parameter sets the value of the clear request retransmission counter. The default value is 1. The

R27 parameter sets the value of the reject retransmission counter. The default value is 0.

The NPVC parameter sets the number of channels reserved as PVCs. The default is 0.

The DEFPCPAR parameter specifies the index of the default call definition to use for this interface. The default value is 0, which means no default call. The call definition (if not 0) must already exist.

The DTEADDRESS parameter specifies the DTE address for the interface.

The LIC parameter specifies the lowest incoming channel number. The default is 0. The HIC parameter specifies the highest incoming channel number. A value of 0 means no incoming channels. The default is 0. The LTC parameter specifies the lowest two-way channel number. The default is 1. The HTC parameter specifies the highest two-way channel number. A value of 0 means no two-way channels. The default is 4095. The LOC parameter specifies the lowest outgoing channel number. The default is 0. The HOC parameter specifies the highest outgoing channel number. A value of 0 means no outgoing channels. The default is 0. The values of LIC, HIC, LTC, HTC, LOC, LTC, NPVC and MAXACTIVE must be consistent with one another.

The ROLE parameter specifies the role this DTE plays in a DTE-DTE environment. The default is DYNAMIC, which means during the restart procedure the two DTEs will negotiate which DTE will acts as DCE.

Examples To configure X.25 DTE interface 1 to negotiate its role in a DTE-to-DTE link, use the command:

```
SET X25T=1 ROLE=DYNAMIC
```

See Also ADD X25T CPAR
CREATE X25T
DESTROY X25T
SET X25T CPAR
RESET X25T
SHOW X25T
SHOW X25T CPAR

SET X25T CPAR

Syntax SET X25T CPAR=*call-index* [MAXDATA={128|256|512|1024}]
[RMAXDATA={128|256|512|1024}] [RWINDOW=1..127]
[TMAXDATA={128|256|512|1024}] [TWINDOW=1..127]
[USERDATA=*hex-string*] [WINDOW=1..127]

where:

- *call-index* is the index of the call parameters to modify.
- *hex-string* is a string of 2 to 20 hexadecimal digits.

Description This command is used to change the attributes of a set of call parameters. The call parameters must already exist, and must not be the default for an X.25 DTE interface or PVC, or be in use by an active call.

The TMAXDATA and RMAXDATA parameters specify the maximum packet sizes for transmission and reception, respectively. The MAXDATA parameter is a shorthand form to specify the same maximum packet size for both transmission and reception. In all cases, the default is 0, which means use the interface default for a call (128).

The TWINDOW and RWINDOW parameters specify the window for transmission and reception, respectively. The WINDOW parameter is a shorthand form to specify the same window for both transmission and reception. In all cases, the default is 0, which means use the interface default for a call (2).

The USERDATA parameter specifies the contents of the user data field for the call request for this call. This is used to specify the protocol that this circuit will carry.

Examples To set the contents of the user data field in all call requests using call parameter set 7 to "ff0d5B3c", use the command:

```
SET X25T CPAR=7 USERDATA=ff0d5b3c
```

See Also ADD X25T CPAR
DELETE X25T CPAR
SHOW X25T CPAR

SHOW MIOX

Syntax SHOW MIOX [=x25t-interface]

where:

- *x25t-interface* is the index number of the X.25 DTE logical interface, in the range 0 to 7.

Description This command displays the operational parameters for the MIOX entity related to the X.25 DTE interface (Figure 5-1 on page 5-23, Table 5-2 on page 5-24). If the X.25 DTE interface is not specified, information for all MIOX entities is displayed.

Figure 5-1: Example output from the SHOW MIOX command.

miox interface 0									

timer values									
minopen	25	inactive	25	holddown	240	collision	60	failure	60
miox interface 1									

timer values									
minopen	60	inactive	60	holddown	60	collision	35	failure	15

Table 5-2: Parameters displayed in the output of the SHOW MIOX command.

Parameter	Meaning
minopen	The value (in seconds) of the minimum open timer.
inactive	The value (in seconds) of the inactivity timer.
holddown	The value (in seconds) of the hold down timer.
collision	The value (in seconds) of the collision retry timer.
failure	The value (in seconds) of the link failure timer.

Examples To display the MIOX configuration for all X.25 DTE interfaces, use the command:

```
SHOW MIOX
```

See Also SET MIOX

SHOW MIOX COUNT

Syntax SHOW MIOX[=*x25t-interface*] COUNT

where:

- *x25t-interface* is the index number of the X.25 DTE logical interface, in the range 0 to 7.

Description This command displays the counters for the MIOX entity related to the X.25 DTE interface (Figure 5-2 on page 5-24, Table 5-3 on page 5-25). If the X.25 DTE interface is not specified, counters for all MIOX entities are displayed.

Figure 5-2: Example output from the SHOW MIOX COUNT command.

miox interface 0					

statistics					
circuits	1				
link state	ALIVE				
incoming calls		failure reason			
accepted	4	bad encap	0		
failed	1	bad protocol	0		
total calls	4	call exists	1		
outgoing calls					
initiation cause		closure cause		failure cause	
closed	8	minopen	0	failure	0
holddown	0	inactive	3	collision	0
collision	1	network	2	network	1
total calls	9	successful	5	unsuccessful	1
packets					
total sent	38	total recvd	42	discarded	0
timeouts					
minopen	9	inactive	3	holddown	1
collision	0	failure	0		

Table 5-3: Parameters displayed in the output of the SHOW MIOX COUNT command.

Parameter	Meaning
miox interface	The number of the X.25 DTE logical interface.
statistics	General MIOX statistics for the DTE interface.
circuits	The total number of MIOX circuits configured for the MIOX entity.
link state	The status of the X.25 link; one of "ALIVE" or "DEAD".
incoming calls	Information about incoming calls on this DTE interface.
accepted	The number of incoming calls that were accepted.
failed	The number of incoming calls that were rejected.
total calls	The total number of incoming calls.
failure reason	Information about the causes of failed incoming calls.
bad encap	The number of incoming calls that were rejected due to an unsupported encapsulation.
bad protocol	The number of incoming calls that were rejected due to a supported encapsulation with an unattached protocol.
call exists	The number of incoming calls that were rejected due to the circuit already having an active connection.
outgoing calls	Information about outgoing calls.
initiation cause	Information about the initiation of outgoing calls.
closed	The number of outgoing calls that were initiated when the circuit was in the closed state.
holddown	The number of calls that were initiated after a previously failed call on a circuit.
collision	The number of calls that were initiated after the previous call failed due to insufficient free X.25 channels to make the call on.
total calls	The total number of outgoing calls initiated.
closure cause	Information about the reasons for closing outgoing calls.
minopen	The number of calls that were closed due to the expiry of the minimum open timer.
inactive	The number of calls that were closed due to the expiry of the call inactivity timer.
network	The number of calls that were closed by the network. This includes calls that were closed by the remote router.
successful	The number of calls that were closed because they were completed successfully.
failure cause	Information about the causes of failed outgoing calls.
failure	The number of calls that failed due to the call not being confirmed within the call failure timer period.
collision	The number of calls that failed due to insufficient free X.25 channels to make the call.
network	The number of calls that failed due to the network or the remote router refusing the call.
unsuccessful	The number of calls that were closed because they were not completed successfully.

Table 5-3: Parameters displayed in the output of the SHOW MIOX COUNT command. (Continued)

Parameter	Meaning
packets	Packet statistics.
total sent	The number of packets sent on the X.25 DTE interface by the MIOX entity.
total recvd	The number of packets received on the X.25 DTE interface by the MIOX entity.
discarded	The number of packets that were received and discarded due to an unsupported protocol.
timeouts	Information about timeouts.
minopen	The number of times the minimum open timer has expired.
inactive	The number of times the inactivity timer has expired.
holddown	The number of times the hold down timer has expired.
collision	The number of times the collision retry timer has expired.
failure	The number of times the link failure timer has expired.

Examples To display the MIOX counters for X.25 DTE interface 1, use the command:

```
SHOW MIOX=1 COUNT
```

See Also SET MIOX

SHOW MIOX CIRCUIT

Syntax `SHOW MIOX[=x25t-interface] CIRCUIT[=circuit-name]
[COUNTER|ENCAP]`

where:

- *x25t-interface* is the index number of the X.25 DTE logical interface, in the range 0 to 7.
- *circuit-name* is an alphanumeric string, 1 to 15 characters in length.

Description This command displays information about MIOX circuits for the MIOX entity related to the specified X.25 DTE interface. If the X.25 DTE interface is not specified, circuits for all MIOX entities are displayed. If a circuit name is specified then only circuits matching the circuit name will be displayed. If neither COUNTER or ENCAP is specified, the operational parameters for the MIOX circuit are displayed (Figure 5-3 on page 5-27, Table 5-4 on page 5-27). If COUNTER is specified counters for the MIOX circuit are displayed (Figure 5-4 on page 5-27, Table 5-5 on page 5-28). If ENCAP is specified encapsulation information and specific counters for higher layer protocols using the circuit are displayed (Figure 5-5 on page 5-29, Table 5-6 on page 5-29).

Figure 5-3: Example output from the SHOW MIOX CIRCUIT command.

miox interface 0						

headoffice		comment				ENABLED
pvc	1	Permanent Circuit to Head Office.				
encap	NULL	cpar	0	comp	OFF	tcpcomp OFF
regionaloffice		comment				ENABLED
dteaddress	300043275					
encap	IP	cpar	0	comp	OFF	tcpcomp ON

Table 5-4: Parameters displayed in the output of the SHOW MIOX CIRCUIT command.

Parameter	Meaning
miox interface	The number of the X.25 DTE logical interface.
<circuit-name>	The name of the MIOX circuit.
comment	A string that describes the circuit.
ENABLED	The status of the circuit; one of "ENABLED" or "DISABLED".
pvc	The PVC for this MIOX circuit (displayed when the circuit is configured to operate over a PVC).
dteaddress	The DTE address for this MIOX circuit (displayed when the circuit is configured to operate over an SVC).
encap	The type of encapsulation for the circuit.
cpar	The call parameter entry to use for the circuit.
comp	Whether or not payload compression is enabled for the circuit.
tcpcomp	Whether or not TCP header compression is enabled for the circuit.

Figure 5-4: Example output from the SHOW MIOX CIRCUIT COUNTER command.

miox interface 0					

regionaloffice					
incoming calls		failure reason			
accepted	10	bad encap	0		
failed	2	bad protocol	2		
total calls	12	call exists	0		
outgoing calls					
initiation cause		closure cause		failure cause	
closed	5	minopen	0	failure	0
holddown	2	inactive	3	collision	0
collision	0	network	1	network	1
total calls	7	successful	4	unsuccessful	1
packets					
total sent	355	total recvd	432	discarded	0
timeouts					
minopen	9	inactive	3	holddown	2
collision	0	failure	0		

Table 5-5: Parameters displayed in the output of the SHOW MIOX CIRCUIT COUNTER command.

Parameter	Meaning
miox interface	The number of the X.25 DTE logical interface.
<circuit-name>	The name of the MIOX circuit.
incoming calls	Information about incoming calls on this DTE interface.
accepted	The number of incoming calls that were accepted for the circuit.
failed	The number of incoming calls that were rejected for the circuit.
total calls	The total number of incoming calls for the circuit.
failure reason	Information about the causes of failed incoming calls.
bad encap	The number of incoming calls that were rejected due the circuit not supporting encapsulation.
bad protocol	The number of incoming calls that were rejected due to a supported encapsulation with an unattached protocol.
call exists	The number of incoming calls that were rejected due to the circuit already having an active connection.
outgoing calls	Information about outgoing calls.
initiation cause	Information about the initiation of outgoing calls.
closed	The number of outgoing calls that were initiated when the circuit was in the closed state.
holddown	The number of calls that were initiated after a previously failed call on a circuit.
collision	The number of calls that were initiated after the previous call failed due to insufficient free X.25 channels to make the call on.
total calls	The total number of outgoing calls for the circuit.
closure cause	Information about the reasons for closing outgoing calls.
minopen	The number of calls that were closed due to the expiry of the minimum open timer.
inactive	The number of calls that were closed due to the expiry of the call inactivity timer.
network	The number of calls that were closed by the network. This includes calls that were closed by the remote router.
successful	The total number of successful outgoing calls for the circuit.
failure cause	Information about the causes of failed outgoing calls.
failure	The number of calls that failed due to the call not being confirmed within the call failure timer period.
collision	The number of calls that failed due to insufficient free X.25 channels to make the call on.
network	The number of calls that failed due to the network or the remote router refusing the call.
unsuccessful	The total number of outgoing calls for the circuit that failed.
packets	Packet statistics.
total sent	The number of packets sent on the MIOX circuit.
total recvd	The number of packets received on the MIOX circuit.
discarded	The number of packets that were received and discarded due to an unsupported protocol.

Table 5-5: Parameters displayed in the output of the SHOW MIOX CIRCUIT COUNTER command. (Continued)

Parameter	Meaning
timeouts	Information about timeouts.
minopen	The number of times the minimum open timer has expired for the circuit.
inactive	The number of times the inactivity timer has expired for the circuit.
holddown	The number of times the hold down timer has expired for the circuit.
collision	The number of times the collision retry timer has expired for the circuit.
failure	The number of times the link failure timer has expired for the circuit.

Figure 5-5: Example output from the SHOW MIOX CIRCUIT ENCAP command.

miox interface 0					

headoffice					
module	IP	packets sent	45	packets rcvd	32
channel	1	state	ACTIVE		
regionaloffice					
module	IP	packets sent	420	packets rcvd	598
channel	245	calls			
state		successful	failed	accepted	
OPEN		12	0	4	

Table 5-6: Parameters displayed in the output of the SHOW MIOX CIRCUIT ENCAP command.

Parameter	Meaning
miox interface	The number of the X.25 DTE logical interface.
<circuit-name>	The name of the MIOX circuit.
module	Attached higher layer protocol.
packets sent	Number of packets sent by the specified protocol.
packets rcvd	Number of packets received for the specified protocol.
channel	Displays the X.25 channel number that the call is currently using.
state	Displays the link state for PVCs and the call state for circuits operating over SVCs.
successful	Displayed for SVCs, shows the number of successful outgoing calls.
failed	Displayed for SVCs, shows the number of failed outgoing calls.
accepted	Displayed for SVCs, shows the number of incoming calls accepted.

Examples To display the operational parameters for MIOX circuit “HeadOffice” on X.25 DTE interface 1, use the command:

```
SHOW MIOX=1 CIRCUIT=HeadOffice
```

To display the counters for MIOX circuit “HeadOffice” on X.25 DTE interface 1, use the command:

```
ACTIVATE MIOX=1 CIRCUIT=HeadOffice COUNTER
```

See Also SET MIOX

SHOW X25T

Syntax SHOW X25T [=x25-interface] [{CIRCUIT|COUNT}]

where:

- *x25-interface* is the number of the X.25 DTE logical interface, in the range 0 to 7.

Description This command displays information about the specified X.25 DTE interface. If the interface is not specified, information about all X.25 DTE interfaces is displayed (Figure 5-6 on page 5-30, Table 5-7 on page 5-31). The CIRCUIT parameter displays details of X.25 circuits (Figure 5-7 on page 5-32, Table 5-8 on page 5-32). The COUNT parameter displays X.25 DTE counters (Figure 5-8 on page 5-33, Table 5-9 on page 5-33).

Figure 5-6: Example output from the SHOW X25T command.

```
-----
X.25 DTE Interface: 0
```

Over:	LAPD0	DTE address:
Packet modulus:	8	Max channels: 128
DTE role:	DYNAMIC	Number PVCs: 499

LIC:	500	LTC:	1500	LOC:	2500
HIC:	1000	HTC:	2000	HOC:	3000
Incoming:	500	Twoway:	500	Outgoing:	500

Timers

Restart	T20:	180	R20:	1	Win Rotate	T25:	60	R25:	0
Call Req	T21:	200			Interrupt	T26:	180		
Reset Req	T22:	180	R22:	1	Reject	T27:	60	R27:	0
Clear Req	T23:	180	R23:	1	Registration	T28:	300	R28:	0
Win Status	T24:	60							

Default call parameters: 0

Receive window:	2	Transmit window:	2
Receive data:	128	Transmit data:	128

```
-----
```


Table 5-7: Parameters displayed in the output of the SHOW X25T command.

Parameter	Meaning
X.25 DTE instance	The number of the X.25 logical interface.
Over	The Layer 2 entity used by this X.25 interface.
DTE address	The DTE address for this X.25 interface.
Packet modulus	The modulus of packet sequence numbers.
Max channels	The maximum number of channels supported.
DTE role	The operational mode of the X.25 interface; one of "DCE", "DTE" or "DYNAMIC".
Number PVCs	The number of channels reserved for PVCs.
LIC	The lowest incoming channel number.
LTC	The lowest two-way channel number.
LOC	The lowest outgoing channel number.
HIC	The highest incoming channel number.
HTC	The highest two-way channel number.
HOC	The highest outgoing channel number.
Incoming	The number of incoming call channels available.
Twoway	The number of twoway call channels available.
Outgoing	The number of outgoing call channels available.
T20	The value (in seconds) of the restart timer.
R20	The value of the restart retransmission counter.
T21	The value (in seconds) of the call timer.
T22	The value (in seconds) of the reset timer.
R22	The value of the restart request retransmission counter.
T23	The value (in seconds) of the clear timer.
R23	The value of the clear request retransmission counter.
T24	The value (in seconds) of the optional window timer.
T25	The value (in seconds) of the window rotation timer.
R25	The value of the window rotation counter.
T26	The value (in seconds) of the interrupt timer.
T27	The value (in seconds) of the reject response timer.
R27	The value of the reject retransmission counter.
T28	The value (in seconds) of the registration timer.
R28	The value of the registration counter.
Default call parameter	The index of the default call parameter set for this interface.
Receive window	The reception window size for the default call parameter set.
Transmit window	The transmission window size for the default call parameter set.
Receive data	The maximum packet size for reception for the default call parameter set.
Transmit data	The maximum packet size for transmission for the default call parameter set.

Figure 5-7: Example output from the SHOW X25T CIRCUIT command.

```

X.25 DTE Instance: 0      Circuits
-----
Channel: 1
Comment: Call to head office

Type:                      PVC
Status:                    pvc
Call Parameters:           0
Established Time:          0
Octets Received:           0      Octets Sent:                0
Data Packets Received:     0      Data Packets Sent:        0
Interrupts Received:       0      Interrupts Sent:         0
Remote Resets:             0      Provider Resets:         0

Timer      T22      T25      T26
Timeouts:  0        0        0
-----

```

Table 5-8: Parameters displayed in the output of the SHOW X25T CIRCUIT command.

Parameter	Meaning
X.25 DTE instance	The number of the X.25 DTE logical interface.
Channel	The channel number.
Comment	A description of the call.
Type	The channel type: one of "PVC" or "SVC".
Status	The status of the call.
CPar	The index of the call parameter set for this channel.
Estab Time	The time that the call was established.
Octets Received	The number of octets received on this channel.
Octets Sent	The number of octets transmitted on this channel.
Data Packets Received	The number of PDUs received on this channel.
Data Packets Sent	The number of PDUs transmitted on this channel.
Interrupts Received	The number of interrupt requests received.
Interrupts Sent	The number of interrupt requests sent.
Remote Resets	The number of calls on this channel reset by this DTE.
Provider Resets	The number of calls on this channel reset by the provider.
T22 timeouts	The number of times a timeout occurred for the reset timer on this channel.
T25 timeouts	The number of times a timeout occurred for the window rotation timer on this channel.
T26 timeouts	The number of times a timeout occurred for the Interrupt timer on this channel.

Figure 5-8: Example output from the SHOW X25T COUNT command.

X.25 DTE instance: 0		Counters	
Data Packets Received:	0	Data Packets Sent:	0
Interrupts Received:	0	Interrupts Sent:	0
Calls Received:	0	Calls Attempted:	0
Calls Refused:	0	Calls Failed:	0
Remote Resets:	0	Provider Resets:	0
Clear Call Received:	0		
Restarts Received:	0		
Protocol Errors:	0		
Incoming Circuits:	0		
Outgoing Circuits:	0		
Twoway Circuits:	0		
Timer Timeouts			
Restart Request T20:	0	Clear Request T23:	0
Call Request T21:	0	Win Rotation T25:	0
Reset Request T22:	0	Interrupt T26:	0
Retrys Exceeded:	0		

Table 5-9: Parameters displayed in the output of the SHOW X25T COUNT command.

Parameter	Meaning
X.25 DTE instance	The number of the X.25 logical interface.
Data Packets Received	The number of data packets received.
Data Packets Sent	The number of data packets transmitted.
Interrupts Received	The number of interrupt requests received.
Interrupts Sent	The number of interrupt requests sent.
Calls Received	The number of incoming calls received.
Calls Attempted	The number of outgoing calls made.
Calls Refused	The number of incoming calls refused.
Calls Failed	The number of outgoing calls the failed.
Remote Resets	The number of times an incoming call was reset by this DTE.
Provider Resets	The number of times an incoming call was reset by the network provider.
Clear Call Received	The number of clear call requests received.
Restarts Received	The number of restart requests received.
Protocol Errors	The number of data packets received which contained protocol errors.
Incoming	The number of incoming call channels available.
Twoway	The number of twoway call channels available.
Outgoing	The number of outgoing call channels available.
T20 timeouts	The number of times a timeout occurred for the restart timer.
T21 timeouts	The number of times a timeout occurred for the call timer.
T22 timeouts	The number of times a timeout occurred for the reset timer.

Table 5-9: Parameters displayed in the output of the SHOW X25T COUNT command. (Continued)

Parameter	Meaning
T23 timeouts	The number of times a timeout occurred for the clear timer.
T25 timeouts	The number of times a timeout occurred for the window rotation timer.
T26 timeouts	The number of times a timeout occurred for the Interrupt timer.
Retrys Exceeded	The number of times the retry counter was exceeded.

Examples To display the counters for X.25 DTE interface 1, use the command:

```
SHOW X25T=0 COUNT
```

See Also CREATE X25T
DESTROY X25T
SET X25T
RESET X25T

SHOW X25T CPAR

Syntax SHOW X25T CPAR[=*call-index*]

where:

- *call-index* is the index of the call parameters to display.

Description This command displays a set of call parameters for X.25 (Figure 5-9 on page 5-34, Table 5-10 on page 5-35). If the call parameter index is specified, that call parameter entry is displayed, otherwise all call parameter entries are displayed. The call parameters must already exist.

Figure 5-9: Example output from the SHOW X25T CPAR command.

```
X.25 DTE call parameters

Index: 1      References: 0
Receive window: 2      Transmit window: 2
Receive data: 128      Transmit data: 128
User data: 80 00 00 00 08 00
NUI string: bt_test_host

Index: 2      References: 3
Receive window: 2      Transmit window: 2
Receive data: 128      Transmit data: 128
User data: 80 00 00 00 81 37
NUI string:
```

Table 5-10: Parameters displayed in the output of the SHOW X25T CPAR command.

Parameter	Meaning
Index	The index of the call parameter set.
References	The number of PVCs and X.25 DTE logical interfaces defined that reference this call plus the number of calls in progress using this call parameter set.
Receive window	The reception window size for this call parameter set.
Transmit window	The transmission window size for this call parameter set.
Receive data	The maximum packet size for reception for this call parameter set.
Transmit data	The maximum packet size for transmission for this call parameter set.
User data	The value to be used in the user data field of a call request for this call parameter set.
NUI string	The Network User Identification to be used for this call parameter set.

Examples To display the configuration of all X.25 DTE call parameter sets, use the command:

```
SHOW X25T CPAR
```

See Also ADD X25T CPAR
CREATE X25T
DELETE X25T CPAR
SET X25T
SET X25T CPAR

Chapter 6

Internet Protocol (IP)

Introduction	6-4
The Internet	6-4
Addressing	6-6
Subnets	6-8
Multihoming	6-9
Address Resolution	6-9
DHCP Client	6-11
ICMP	6-11
Routing	6-12
Routing Information Filters	6-13
RIP	6-14
Metrics	6-16
Policy-Based Routing	6-16
Priority-Based Routing	6-18
Route Templates	6-19
Named Hosts	6-20
DNS Relay Agent	6-21
Traffic Filters	6-21
SNMP	6-23
Control and Debug Commands	6-24
Ping and Trace Route	6-25
Security Options	6-26
Broadcast Forwarding	6-26
Examples	6-27
BOOTP Relay Agent	6-29
IP Multicasting	6-31
Remote Address Assignment	6-32
IP Address Pools	6-32
Configuration Examples	6-33
A Basic TCP/IP Setup	6-33
Troubleshooting	6-37
Configuring IP Filters	6-38
Command Reference	6-43
ADD BOOTP RELAY	6-43
ADD IP ARP	6-44
ADD IP FILTER	6-45
ADD IP HELPER	6-51
ADD IP HOST	6-52
ADD IP INTERFACE	6-53
ADD IP RIP	6-56
ADD IP ROUTE	6-57

ADD IP ROUTE FILTER	6-59
ADD IP ROUTE TEMPLATE	6-60
ADD IP TRUSTED	6-61
CREATE IP POOL	6-62
DELETE BOOTP RELAY	6-63
DELETE IP ARP	6-63
DELETE IP FILTER	6-64
DELETE IP HELPER	6-64
DELETE IP HOST	6-65
DELETE IP INTERFACE	6-66
DELETE IP RIP	6-67
DELETE IP ROUTE	6-68
DELETE IP ROUTE FILTER	6-69
DELETE IP ROUTE TEMPLATE	6-69
DELETE IP TRUSTED	6-69
DELETE TCP	6-70
DESTROY IP POOL	6-71
DISABLE BOOTP RELAY	6-71
DISABLE IP	6-72
DISABLE IP DEBUG	6-72
DISABLE IP DNSRELAY	6-73
DISABLE IP ECHOREPLY	6-73
DISABLE IP FOFILTER	6-73
DISABLE IP FORWARDING	6-74
DISABLE IP HELPER	6-74
DISABLE IP INTERFACE	6-75
DISABLE IP REMOTEASSIGN	6-75
DISABLE IP ROUTE	6-76
DISABLE IP SRCROUTE	6-76
ENABLE BOOTP RELAY	6-77
ENABLE IP	6-77
ENABLE IP DEBUG	6-78
ENABLE IP DNSRELAY	6-78
ENABLE IP ECHOREPLY	6-78
ENABLE IP FOFILTER	6-78
ENABLE IP FORWARDING	6-79
ENABLE IP HELPER	6-80
ENABLE IP INTERFACE	6-80
ENABLE IP REMOTEASSIGN	6-81
ENABLE IP ROUTE	6-81
ENABLE IP SRCROUTE	6-82
PING	6-82
PURGE BOOTP RELAY	6-84
PURGE IP	6-84
RESET IP	6-84
RESET IP COUNTER	6-85
RESET IP INTERFACE	6-85
SET BOOTP MAXHOPS	6-86
SET IP ARP	6-86
SET IP AUTONOMOUS	6-87
SET IP FILTER	6-88
SET IP HOST	6-91
SET IP INTERFACE	6-92
SET IP LOCAL	6-94
SET IP NAMESERVER	6-95
SET IP RIP	6-96
SET IP RIPTIMER	6-97
SET IP ROUTE	6-98
SET IP ROUTE FILTER	6-100

SET IP ROUTE TEMPLATE	6-101
SET IP SECONDARYNAMESEVER	6-102
SET PING	6-103
SET TRACE	6-104
SHOW BOOTP RELAY	6-105
SHOW IP	6-106
SHOW IP ARP	6-108
SHOW IP COUNTER	6-109
SHOW IP DEBUG	6-116
SHOW IP FILTER	6-117
SHOW IP HELPER	6-119
SHOW IP HOST	6-120
SHOW IP INTERFACE	6-121
SHOW IP POOL	6-124
SHOW IP RIP	6-126
SHOW IP RIPTIMER	6-127
SHOW IP RIP COUNTER	6-128
SHOW IP ROUTE	6-130
SHOW IP ROUTE FILTER	6-134
SHOW IP ROUTE TEMPLATE	6-135
SHOW IP TRUSTED	6-136
SHOW IP UDP	6-136
SHOW PING	6-137
SHOW TCP	6-139
SHOW TRACE	6-143
STOP PING	6-145
STOP TRACE	6-145
TRACE	6-146

Introduction

This chapter describes the main features of the Internet Protocol (IP), support for IP on the router, and how to configure and operate the router to route IP protocols.

The router is capable of routing IP data packets via the wide area network. This allows a group of remote LANs to be joined together as a single IP autonomous system and to be connected to other IP networks such as the Internet.

IP protocols are widely used and available on nearly every host and PC system. They provide a range of services including remote login, file transfer and Email. Using IP routers allows these services to be fully supported within an organisation and to other organisations internationally.



IP is often referred to as TCP/IP. The letters TCP refer to Transmission Control Protocol. This is a protocol which runs over IP and provides end-to-end reliability and control of IP network connections. A closely related protocol called UDP (User Datagram Protocol) also runs over IP and is used in situations where reliable transport of datagrams is not required. Both TCP and UDP are used by modules in the router. TCP is used to implement Telnet remote logins, while UDP is used for downloading software.

The Internet

The Internet (with a capital “I”) is the name given to the large, worldwide network of networks based on the original concepts of the ARPAnet. A large number of government, academic and commercial organisations are connected to the Internet, and use it to exchange traffic such as Email. The Internet uses the TCP/IP protocols for all routing. In recent times the term internet (with a lowercase “i”) has also come to refer to any network (usually a wide area network) which utilises the Internet Protocol. The remainder of this chapter will concentrate on the latter definition, i.e. that of a generalised network which uses IP as the transport protocol.

The basic unit of data sent through an internet is a *packet* or *datagram*. An IP network functions by moving packets between routers and/or hosts. A packet consists of a *header* followed by the *data* (Figure 6-1 on page 6-5, Table 6-1 on page 6-5). The header contains the information necessary to move the packet across the internet. It must be able to cope with missing and duplicated packets as well as possible fragmentation (and reassembly) of the original packet.

Packets are sent using a *connectionless* transport mechanism. A connection is not maintained between the source and destination addresses; rather, the destination address is placed in the header and the packet is transmitted on a best effort basis. It is up to the intermediate systems (routers and gateways) to deliver the packet to the correct address, using the information in the header. Successive packets may take different routes through the network to the destination. There is a strong analogy with the postal delivery system in that letters are placed in individually addressed envelopes and put into the system in the ‘hope’ that they will arrive. Like an internet, the postal system is very reliable. In an internet, higher layers (such as TCP and Telnet) are responsible for ensuring that packets are delivered in a reliable and sequenced way.

In contrast to a connectionless transport mechanism, a *connection-oriented* transport mechanism requires a connection to be maintained between the source and destination for as long as necessary to complete the exchange of packets between source and destination. X.25 is an example of a connection-oriented protocol. A good analogy to X.25 would be a telephone call, in which both parties verify that they are talking to the correct person before exchanging highly sequenced data (if both talk at once then nothing intelligible results!), and the connection is maintained until both parties have finished talking. Its not hard to imagine the chaos if the telephone system delivered words in the wrong order.

Figure 6-1: Format of an IP datagram.

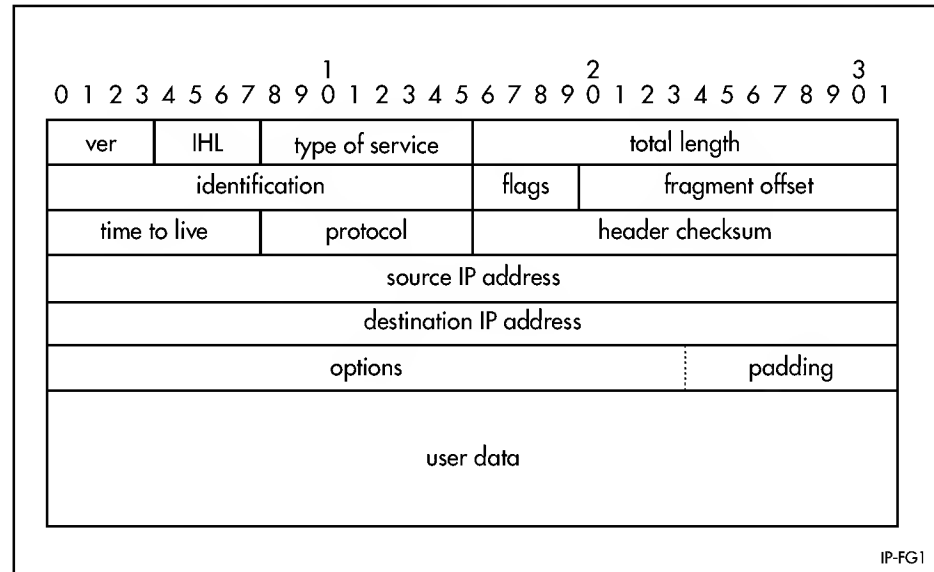


Table 6-1: Functions of the fields in an IP datagram.

Field	Function
ver	The version of the IP protocol that created the datagram.
IHL	The length of the IP header in 32-bit words (the minimum value is 5).
Type of service	The quality of service (precedence, delay, throughput, and reliability) desired for the datagram.
Total length	The length of the datagram (both header and user data), in octets.
Identification	A 16-bit value assigned by the originator of the datagram, used during reassembly.
Flags	Control bits indicating whether the datagram may be fragmented, and if so, whether other later fragments exist.
Fragment offset	The offset in the original datagram of the data being carried in this datagram, for fragmented datagrams.
Time to live	The time in seconds the datagram is allowed to remain in the internet system.
Protocol	The high level protocol used to create the message (analogous to the type field in an Ethernet packet).
Header checksum	A checksum of the header.
Source IP address	32-bit IP address of the sender.

Table 6-1: Functions of the fields in an IP datagram. (Continued)

Field	Function
Destination IP address	32-bit IP address of the recipient.
Options	An optional field primarily used for network testing or debugging.
Padding	All bits set to zero—used to pad the datagram header to a length that is a multiple of 32 bits.
User data	The actual data being sent.

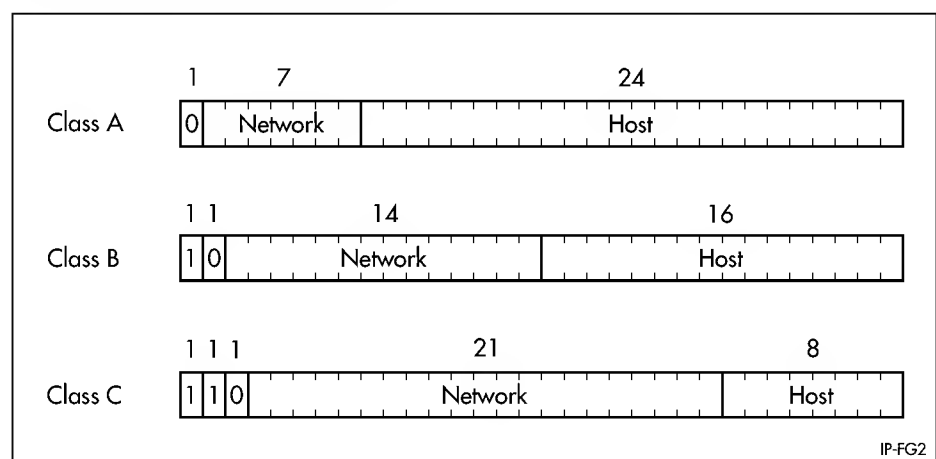
Addressing

Internet addresses are fundamental to the operation of the TCP/IP internet. Each packet must contain an internet address to determine where to send the packet. Most packets also require a source address so that the sender of the packet is known. Addresses are 32-bit quantities which are logically divided into fields. They must not be confused with physical addresses (such as an Ethernet address); they serve only to address Internet Protocol packets. Addresses are organised into five classes (Table 6-2 on page 6-6).

Table 6-2: Internet Protocol address classes and limits on numbers of networks and hosts.

Class	Maximum number of possible networks	Maximum number of hosts per network
A	127	16,777,216
B	16,384	65,536
C	2,097,152	255
D	Reserved Class	
E	Reserved Class	

Each class differs in the number of bits assigned to the host and network portions of the address (Figure 6-2 on page 6-6).

Figure 6-2: Subdivision of the 32 bits of an Internet address into network and host fields for class A, B and C networks.

The addressing scheme is designed to allow routers to efficiently extract the host and network portions of an address. In general a router is only interested in the network portion of an address.

Class A sets the *Most Significant Bit* (MSB) to 0 and allocates the next 7 bits to define the network and the remaining 24 bits to define the host. Class B sets the two MSBs to 10 and allocates the next 14 bits to designate the network while the remaining 16 refer to the host. Class C sets the three MSBs to '110' and allocates the next 21 bits to designate the network while the remaining 8 are left to the user to assign as host or subnet numbers.

The term host refers to any attached device on a subnet, including PCs, mainframes and routers. Most hosts are connected to only one network. In other words they have a single IP address. Routers are connected to more than one network and can have multiple IP addresses. The IP address is expressed in *dotted decimal notation* by taking the 32 binary bits and forming 4 groups of 8 bits, each separated by a dot. For example:

```
10.4.8.2 is a class A address
10 is the DDN assigned network number
.4.8 are (possibly) user assigned subnet numbers
.2 is the user assigned host number
```

```
172.16.9.190 is a class B address
172.16 is the DDN assigned network number
.9 is the user assigned subnet number
.190 is the user assigned host number
```

The value 0.0.0.0 is used to define the default address, while a value of all ones in any host portion (i.e. 255) is reserved as the broadcast address. Some older versions of UNIX use a broadcast value of all zeros, therefore both the value '0' and the value '255' are reserved within any user assigned host portion. The address 172.16.0.0 refers to **any** host (not **every** host) on **any** subnet within the class B address 172.16. Similarly 172.16.9.0 refers to **any** host on subnet 9, whereas 172.16.9.255 is a packet addressed to **every** host on subnet 9. The router uses this terminology to indicate where packets are to be sent.

An address with '0' in the host portion refers to 'this particular host' while an address with '0' in the network portion refers to 'this particular network'. As mentioned above a value of all '1' (255) is a broadcast. To reduce loading, IP consciously tries to limit broadcasts to the smallest possible set of hosts, hence most broadcasts are 'directed'. For example 172.16.56.255 is a broadcast to subnet 56 of network 172.16.

A major problem with the IP type of addressing is that it defines connections not hosts. A particular address, although it is unique, defines a host by its connection to a particular network. Therefore if the host is moved to another network the address must also change. The situation is analogous to the postal system. A related problem can occur when an organisation which has a class C address finds that they need to upgrade to class B. This involves a total change of every address for all hosts and routers. Thus the addressing system is not scalable.

Subnets

Related to the two issues discussed above, the rapid growth of the Internet has meant a proliferation in the number of addresses which must be handled by the core routers. More addresses means more loading and tends to slow the system down. This is overcome by minimising the number of network addresses by sharing the same IP prefix (the assigned network number) with multiple physical networks. Generally these would all be within the same organisation, although this is not a requirement. There are two main ways of achieving this; Proxy ARP and subnetting. Proxy ARP will be discussed later in this section.

A subnet is formed by taking the host portion of the assigned address and dividing it into two parts. The first part is the 'set of subnets' while the second refers to the hosts on **each** subnet. For example the DDN may assign a class B address as 172.16.0.0. The system manager would then assign the lower two octets in some way which makes sense for this particular network. A common method for class B is to simply use the higher octet to refer to the subnet. Thus there are 254 subnets (0 and 255 are reserved) each with 254 hosts. These subnets need not be physically on the same media. Generally they would be allocated geographically with subnet 2 being one site, subnet 3 another and so on. Some sites may have a requirement for multiple subnets on the same LAN. This could be to increase the number of hosts or simply to make administration easier. In this case it is normal (but not required) that the subnets be assigned contiguously for this site. This makes the allocation of a subnet mask easier. This mask is needed by the routers to ascertain which subnets are available at each site. Bits in the mask are set to '1' if the router is to treat the corresponding bit in the IP address as belonging to the network portion or set to '0' if it belongs to the host portion. This allows a simple bit-wise logical AND to determine if the address should be forwarded or not. Although the standard does not require that the subnet mask must select contiguous bits, it is normal practice to do so. To do otherwise can make the allocation of numbers rather difficult and prone to errors. Some example masks are:

```
11111111.11111111.11111111.00000000 = 255.255.255.0
<----network----> <subnet> <-host->
```

This would give 254 subnets on a class B network, each with 254 hosts.

```
11111111.11111111.11111111.11110000 = 255.255.255.240
<----network----> <--subnet--><host>
```

This would give 4094 subnets on a class B network, each with 14 hosts or, 14 subnets on a class C network each with 14 hosts.

The official description of subnetting is given in RFC 950. Subnet information and IP addresses are added to the router using the commands:

```
ADD IP INTERFACE=interface IPADDRESS={ipadd|DHCP}
[BROADCAST={0|1}] [DIRECTEDBROADCAST={YES|NO|ON|OFF}]
[FILTER={0..99|NONE}] [FRAGMENT={YES|NO}] [MASK=ipadd]
[METRIC=1..16] [MULTICAST={OFF|SEND|RECEIVE|BOTH|ON}]
[POLICYFILTER={100..199|NONE}] [PRIORITYFILTER={200..299|
NONE}] [PROXYARP={ON|OFF}] [RIPMETRIC=1..16] [VJC={ON|
OFF}]
```

```
SET IP INTERFACE=interface [BROADCAST={0|1}]
    [DIRECTEDBROADCAST={YES|NO|ON|OFF}] [FILTER={0..99|NONE}]
    [FRAGMENT={YES|NO}] [IPADDRESS=ipadd|DHCP] [MASK=ipadd]
    [METRIC=1..16] [MULTICAST={OFF|SEND|RECEIVE|BOTH|ON}]
    [POLICYFILTER={100..199|NONE}] [PRIORITYFILTER={200..299|
    NONE}] [PROXYARP={ON|OFF}] [RIPMETRIC=1..16] [VJC={ON|
    OFF}]
```

Multihoming

The router can be configured as a *multihomed* device with multiple IP addresses. Up to 16 logical IP interfaces can be added to a single layer 2 interface such as Eth0 or PPP0, and up to a total of 512 logical interfaces per router.

An IP interface name is formed by concatenating a layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number in the range 0 to 15. For example, eth0-0 is the first logical IP interface assigned to the Ethernet 0 interface and ppp1-8 is the ninth logical IP interface assigned to the PPP1 interface. If a logical interface is not specified, 0 is assumed. For example, 'eth0' is equivalent to 'eth0-0'.

Each logical interface has its own unique IP address and mask, and can be assigned its own traffic filter, policy filter, and priority filter. Each logical interface has its own interface counters and can be enabled, disabled or reset independently of other logical interfaces assigned to the same layer 2 interface. Each additional logical interface created on a layer 2 interface will add an extra entry to the IP Address Table in the SNMP MIB-II MIB. See *Appendix C, SNMP MIBs* for a complete description of the objects in MIB-II.



The router does not support a single logical interface being associated with multiple physical interfaces in order to increase the reliability or throughput between directly connected machines by providing alternative physical paths between them. This functionality is provided by layer 2 'multiplexing' schemes such as PPP multilink.

Address Resolution

As well as the assigned IP address, most hosts also have a media-dependant physical address. For Ethernet LANs this is a 6-byte globally unique number. In order to communicate, hosts need to know the physical address. The Address Resolution Protocol (ARP) allows the host to find the targets' physical address on the same media, simply by knowing its IP address. It does this by sending out an ARP broadcast packet containing both the source and destination IP address. The broadcast is media-dependant. For Ethernet LANs the broadcast address is a packet whose destination address bits are all '1'. All stations on the LAN receive this address, but only one host is able to recognise its own IP address. It replies, thereby giving the original host its physical address. The ARP protocol is defined in RFC 826 and is a simple but effective use of directed broadcasts. To reduce the number of broadcasts, each host generally keeps a cache of the IP address to physical address mappings (also called bindings). This cache is searched first before a broadcast is attempted to see if a mapping already exists. The ARP cache entries are *aged* to eliminate non-current connections. In the case of a packet destined for a local host, an

initial ARP request is sent. If a response is not received, the ARP request is retried before an ICMP message is sent back to the packets sender.

A static entry can be added to the ARP cache to map hosts which don't support the ARP protocol using the command:

```
ADD IP ARP=ipadd INTERFACE=interface {CIRCUIT=miox-circuit |  
ETHERNET=macadd}
```

Existing static ARP entries can be modified or deleted with the commands:

```
SET IP ARP=ipadd INTERFACE=interface {CIRCUIT=miox-circuit |  
ETHERNET=macadd}  
DELETE IP ARP=ipadd
```

The command:

```
SHOW IP ARP
```

displays the current contents of the router's ARP cache.



It is rarely necessary to add an ARP entry in this way.

The dynamic ARP entries are *aged* to ensure that the table does not fill with entries for hosts which are no longer active. Old entries are deleted. Static ARP entries are not aged.

The router uses a technique called *Proxy ARP* (defined in RFC 1027) to allow hosts which do not support routing (i.e. they have no knowledge of the network structure) to determine the physical addresses of hosts on other networks. The router intercepts ARP broadcast packets and substitutes its own physical address for that of the remote host. This only occurs if the router has the *best* route to the remote host. By responding to the ARP request the router ensures that all subsequent packets from the local host will be directed to the router's physical address and it can then forward these to the remote host. The process is symmetrical. Proxy ARP is enabled by default for each Ethernet interface. It can be disabled selectively using the command:

```
SET IP INTERFACE=interface PROXY=OFF
```

The command:

```
SHOW IP INTERFACE
```

displays details of the interfaces assigned to the IP module, including whether or not Proxy ARP is enabled on each interface.

DHCP Client

IP interfaces can be configured either with a static IP address, or with a dynamic IP address assigned by DHCP (*Dynamic Host Configuration Protocol*). To configure an IP interface to use an address assigned by DHCP, set the IPADDRESS parameter of the interface to DHCP:

```
ADD IP INTERFACE=interface IPADDRESS={ipadd|DHCP}
[BROADCAST={0|1}] [DIRECTEDBROADCAST={YES|NO|ON|OFF}]
[FILTER={0..99|NONE}] [FRAGMENT={YES|NO}] [MASK=ipadd]
[METRIC=1..16] [MULTICAST={OFF|SEND|RECEIVE|BOTH|ON}]
[POLICYFILTER={100..199|NONE}] [PRIORITYFILTER={200..299|
NONE}] [PROXYARP={ON|OFF}] [RIPMETRIC=1..16] [VJC={ON|
OFF}]

SET IP INTERFACE=interface [BROADCAST={0|1}]
[DIRECTEDBROADCAST={YES|NO|ON|OFF}] [FILTER={0..99|NONE}]
[FRAGMENT={YES|NO}] [IPADDRESS={ipadd|DHCP}] [MASK=ipadd]
[METRIC=1..16] [MULTICAST={OFF|SEND|RECEIVE|BOTH|ON}]
[POLICYFILTER={100..199|NONE}] [PRIORITYFILTER={200..299|
NONE}] [PROXYARP={ON|OFF}] [RIPMETRIC=1..16] [VJC={ON|
OFF}]
```

When the IPADDRESS parameter of an IP interface is set to DHCP rather than a static IP address, the router's DHCP client will obtain the IP address and subnet mask for the interface, and other IP configuration parameters, from a DHCP server. See the description of the ADD IP INTERFACE command on page 6-53 for a list of the DHCP reply parameters used by the router to configure IP interfaces.

For example, to configure interface eth0 to obtain its IP address and subnet mask from DHCP, use the command

```
SET IP INTERFACE=eth0 IPADDRESS=DHCP
```

If an IP interface is configured to obtain use its IP address and subnet mask from DHCP, the interface will not take part in IP routing until the IP address and subnet mask have been set by DHCP.



Remote address assignment must be enabled using the ENABLE IP REMOTEASSIGN command before IP interfaces will accept addresses dynamically assigned by DHCP.

ICMP

The *Internet Control Message Protocol* (ICMP) allows routers to send error and control messages to other routers or hosts. It provides the communication between IP software on one system and IP software on another. The router implements all non-obsolete ICMP functions (Table 6-3 on page 6-12). Some early systems may not fully implement all ICMP types. In particular type 11 (Time To Live Exceeded) is frequently not fully implemented.

Table 6-3: ICMP messages implemented by the router.

ICMP packet (type)	Router response
Echo reply (0)	This is used to implement the 'ping' command common to most UNIX and TCP implementations. The router sends out an 'Echo reply' packet in response to a 'Echo request'.
Destination unreachable (3)	This message is sent out when the router drops a packet because it did not have a route to the destination.
Source Quench (4)	The router will send a 'Source Quench' if it must drop a packet due to limited internal resources. This could be because the source was sending data too fast to be forwarded.
Redirect (5)	The router will issue a 'redirect' packet to inform a local host that its target is located on the same LAN (no routing is required) or when it detects a host using a non-optimal route (usually because a link has failed or changed its status).
Echo request (8)	This is related to (1) and results in an 'echo reply' packet being sent. The router can also generate an 'echo request' packet as a result of the PING command on page 6-82.
Time to Live Exceeded (11)	If the TTL field in a packet falls to zero the router will send a 'Time to live exceeded' packet. This could occur if a route was excessively long or if too many hops were in the path.

Routing

The process of routing packets consists of selectively forwarding data packets from one network to another. The router bases its decision to send the packet to a particular network on the information it can learn dynamically from listening to the selected route protocol, as well as the static information entered as part of the configuration process. In addition, user-defined filters can be used to restrict the way packets are sent.

The router maintains a table of routes which tells the router how to find a remote network or host. The route table holds information about routes to destinations. A route is uniquely identified by IP address, network mask, next hop, ifIndex, protocol and policy. A list of routes comprises all the different routes to a destination. The routes may have different metrics, next hops, policy or protocol. A list of routes is uniquely identified by its IP address and net mask.

When an IP packet is received, the routing table is scanned to find the lowest metric route to the destination. Provided that no filters are active which would exclude the packet, it is then forwarded to that route by sending it to the router specified by the next hop. If no route exists the table is scanned for the default route (0.0.0.0) and forwarded as before. If no direct route or default route exists, the packet is discarded and an ICMP message to that effect is sent back to the source.

The routing table is maintained dynamically by using RIP. RIP acts to exchange routing information with other routers or hosts. Routes can also be entered *statically* using the command:

```
ADD IP ROUTE=ipadd INTERFACE=interface NEXTHOP=ipadd
[CIRCUIT=miox-circuit] [MASK=ipadd] [METRIC=1..16]
[METRIC1=1..16] [METRIC2=1..65535] [POLICY=0..7]
[PREFERENCE=0..65535]
```

This is done in two situations:

- To define the 'default' route (0.0.0.0). This is normally used to direct packets for which no learned (or static) routes exist. It would then point to some 'external' network such as the Internet.
- To set up multiple networks or subnets. In this case multiple routes are defined for a particular interface (usually a LAN port). This allows a single physical media to support multiple subnets.

Existing static routes can be modified with the command:

```
SET IP ROUTE=ipadd INTERFACE=interface MASK=ipadd
NEXTHOP=ipadd [CIRCUIT=miox-circuit] [METRIC=1..16]
[METRIC1=1..16] [METRIC2=1..65535] [POLICY=0..7]
[PREFERENCE=0..65535]
```

or deleted altogether with the command:

```
DELETE IP ROUTE=ipadd INTERFACE=interface MASK=ipadd
NEXTHOP=ipadd
```

The command:

```
SHOW IP ROUTE
```

displays the entire routing table, including both static and dynamic routes.

Routes may be cached to improve route lookup performance. The route cache holds the most recently used routes. The route cache can be enabled or disabled using the commands:

```
ENABLE IP ROUTE CACHE
DISABLE IP ROUTE CACHE
```

and is enabled by default. When determining the best route to a destination the cache is searched first, using a hash function calculated from the destination information. If a route is not found in the cache, the entire route table is searched. If a route is found in the route table it is added to the cache. The current contents of the route cache can be displayed using the command:

```
SHOW IP ROUTE CACHE
```

Routing Information Filters

Two mechanisms are provided to manage the process of learning dynamic routes via routing protocols.

Route filters control which routes are received and sent by each routing protocol, over each interface and to particular destinations. When routing information is received by the router, routes that match a filter are added to or omitted from the route table depending on the action defined for the route filter. When the router transmits routing information, routes that match a route filter are included or excluded from the transmission depending on the action

defined for the route filter. A route filter is created or destroyed using the commands:

```
ADD IP ROUTE FILTER [=filter-id] IP=ipadd MASK=ipadd
ACTION={INCLUDE|EXCLUDE} [DIRECTION={RECEIVE|SEND|BOTH}]
[INTERFACE=interface] [NEXTHOP=ipadd] [POLICY=0..7]
[PROTOCOL={ANY|RIP|STATIC|INTERFACE}]
DELETE IP ROUTE FILTER=filter-id
```

The *filter-id* specifies the position in the list of filters. The list of route filters is searched in order until a match is found. The IP, MASK, INTERFACE, NEXTHOP, POLICY and PROTOCOL parameters define a pattern to match against routes. The DIRECTION parameter determines whether the filter applies to route information received, transmitted or both. The ACTION parameter determines whether routes matching the pattern are used or discarded. A route filter is modified using the command:

```
SET IP ROUTE FILTER=filter-id IP=ipadd MASK=ipadd
ACTION={INCLUDE|EXCLUDE} [DIRECTION={RECEIVE|SEND|BOTH}]
[INTERFACE=interface] [NEXTHOP=ipadd] [POLICY=0..7]
[PROTOCOL={ANY|RIP|STATIC|INTERFACE}]
```

The current list of route filters is displayed using the command:

```
SHOW IP ROUTE FILTER
```

The alternative mechanism is to defined one or more trusted routers. A *trusted router* is a source of RIP broadcasts that can be “trusted” to provide up-to-date, valid routing information. If one or more trusted routers are defined, only routing information from the specified source(s) will be accepted by the router and included in the routing table. If no trusted routers are defined, routing information is accepted from any source, although RIP packets may be filtered (e.g. with the ADD IP FILTER command on page 6-45 or the ADD IP ROUTE FILTER command on page 6-59) before reaching the RIP process. A trusted router is added or deleted with the commands:

```
ADD IP TRUSTED=ipadd
DELETE IP TRUSTED=ipadd
```

The list of trusted routers can be displayed with the command:

```
SHOW IP TRUSTED
```

RIP

Routing Information Protocol (RIP) is described fully in RFC 1058. Extensions for RIP version 2 are described in RFC 1723. Extensions for RIP on demand is described in RFC 1582. RIP is a fairly simple distance vector protocol which defines networks based on how many hops they are from the router. Once a network is more than 15 hops away (one hop is one link) it is not included in the routing table.

The possible routes (there may be more than one) to a particular host are selected on the basis of the shortest one. If two routes have the same metric (hop count) or cost the first one found will be chosen. RIP does not cope very well with a meshed (multiply connected) network. It suits star topologies very well.



RIP can have multiple links to a particular destination. It will choose the best one based simply on the metric, which for RIP, is either administratively assigned, or is the hop

count (i.e. number of links). RIP can not send data over multiple paths to a destination. Once a route is chosen, all data is sent over this path until the metric changes.

Each router configured for RIP maintains a relatively simple route table as described earlier. The router will periodically broadcast its routing information to other routers. Similarly it will need to obtain this information from neighbouring routers to improve its own picture of the network. Routes are removed from the table if they are not kept up to date (refreshed) by the neighbouring routers.

The RIP version 2 extensions allow the RIP updates to contain subnet masks and next hop information. The ability to carry subnet masks allows the use of different sized subnet masks on different subnets within the same network.

The RIP on demand extensions allow RIP to be used over demand links that are activated only when there is traffic to send. Route information is only exchanged when there is a change in the routing table and routes obtained over the link are not aged.

RIP broadcasts are automatically enabled when at least one RIP neighbour is defined. RIP neighbours are defined with the command:

```
ADD IP RIP INT=interface
```

The operation of RIP is controlled by four timers whose values are set globally using the command:

```
SET IP RIPTIMER UPDATE=time INVALID=time HOLDDOWN=time  
FLUSH=time
```

The UPDATE parameter sets the time interval between RIP updates for all interfaces not using RIP on demand. The default is to send an update every 30 seconds. The INVALID parameter sets the time interval after which the router will deem a route to be invalid if no update has been received for the route. The default is 180 seconds. The HOLDDOWN parameter sets the time interval, after a route has become invalid, during which the router will ignore updates for the route which would normally make the route valid again. The default is 120 seconds. The FLUSH parameter sets the time interval, from the last update of a route, until the route is flushed from the route table. This time should be set higher than the sum of the INVALID and HOLDDOWN times, and defaults to 300 seconds.

After a valid update, the FLUSH and INVALID timers are restarted. When the INVALID timer expires the route is invalidated and the HOLDDOWN timer started. The FLUSH timer continues to run. When the HOLDDOWN timer expires valid updates for the route will result in the router being reinstated. When the FLUSH timer expires, the route is deleted from the route table.

The current values of the RIP timers can be displayed using the command:

```
SHOW IP RIPTIMER
```

RIP neighbours are removed with the command:

```
DELETE IP RIP INT=interface
```

If no RIP neighbours are defined, RIP broadcasts are disabled. The command:

```
SHOW IP RIP
```

displays the neighbours to which the router is sending RIP broadcasts.

Metrics

Metrics are used to determine the criteria for using one route over another route. In this sense they *measure* some aspect of the route. For RIP the metric is simply the *hop count*, which is a measure of the number of links it will take to get to the specified destination, or in other words how far away it is.

Policy-Based Routing

Destination routing protocols such as RIP use metrics to determine the shortest or optimal path to the destination. Packets are routed via the shortest or optimal path as determined by the route metric. Policy routing is an alternative mechanism for routing packets, based on policies or rules set by the network manager. Policy routing is used in situations where it is desirable for certain packets to be routed some way other than the obvious shortest path, such as to provide equal access, protocol-sensitive routing, source-sensitive routing, routing based on interactive versus batch traffic, or routing based on dedicated links.

The Type of Service (TOS) octet in the IP header comprises three fields: precedence (bits 0 to 2), TOS (bits 3 to 6) and MBZ (bit 7). The precedence field is intended to denote the importance or priority of the datagram, but is not commonly used. The MBZ field should always be zero (0) and is currently unused. The TOS field denotes the type of service required and is used by the network to make trade-offs between throughput, delay, reliability and cost. The TOS field is treated as an integer value between 0 and 15. RFC 1349 defines the semantics of five specific TOS values (Table 6-4 on page 6-16).

Table 6-4: TOS values defined by RFC 1349.

Decimal	Binary	Meaning
8	1000	Minimise delay
4	0100	Maximise throughput
2	0010	Maximise reliability
1	0001	Minimise cost
0	0000	Normal service

Although the semantics of the other values are undefined, they are still legal TOS values and network devices must not prevent the use of such values in any way.

TOS values may be considered when determining the route to use for an IP packet. All routes have an assigned TOS value. This is normally the default TOS value (0), unless the route has been learned using a routing protocol that supports TOS, or the TOS value has been statically assigned.

To forward an IP packet, a router uses the packets destination address to search its route of forwarding table for a route to the destination. If a route is not found, or if the selected route has an infinite metric, the destination is considered unreachable and the packet is discarded. If a single route is found with a finite metric, it is used. If more than one route is found with a finite metric, the TOS values of the selected routes can be used to refine the selection.

A route with a TOS value identical to the TOS value in the IP packet will be used in preference to a route with the default TOS value (0).

The router uses the TOS field in IP routes to implement policy-based routing of IP packets. However, since the TOS field in IP packets is not set or used by many IP implementations, the router makes use of filters to assign the TOS values used for policy routing to IP packets as they are received.

To enable policy routing, the first step is to create a filter to select the IP packets which are to be routed according to policy rather than destination, using a variant of the ADD IP FILTER command on page 6-45:

```
ADD IP FILTER=filter-number SOURCE=ipadd [SMASK=ipadd]
[SPORT={port}] [DESTINATION=ipadd [DMASK=ipadd]]
[DPOR={port}] [ICMPCODE={icmp-code}]
[ICMPTYPE={icmp-type}] [LOG={4..1600|DUMP|HEADER|NONE}]
[OPTIONS={YES|NO}] [PROTOCOL={protocol|ANY|EGP|ICMP|OSPF|
TCP|UDP}] [SESSION={ANY|ESTABLISHED|START}] [SIZE=size]
[ENTRY=entry-number] {ACTION={INCLUDE|EXCLUDE}|
POLICY=0..15|PRIORITY=P0..P7}
```

IP filters with filter numbers in the range 100 to 199 are treated as policy filters. The ACTION parameter is replaced by the POLICY parameter which identifies the policy used to route IP packets which match the filter entry.

The policy filter is then assigned to an interface using the ADD IP INTERFACE command on page 6-53 or the SET IP INTERFACE command on page 6-92:

```
ADD IP INTERFACE=interface IPADDRESS={ipadd|DHCP}
[BROADCAST={0|1}] [DIRECTEDBROADCAST={YES|NO|ON|OFF}]
[FILTER={0..99|NONE}] [FRAGMENT={YES|NO}] [MASK=ipadd]
[METRIC=1..16] [MULTICAST={OFF|SEND|RECEIVE|BOTH|ON}]
[POLICYFILTER={100..199|NONE}] [PRIORITYFILTER={200..299|
NONE}] [PROXYARP={ON|OFF}] [RIPMETRIC=1..16] [VJC={ON|
OFF}]

SET IP INTERFACE=interface [BROADCAST={0|1}]
[DIRECTEDBROADCAST={YES|NO|ON|OFF}] [FILTER={0..99|NONE}]
[FRAGMENT={YES|NO}] [IPADDRESS=ipadd|DHCP] [MASK=ipadd]
[METRIC=1..16] [MULTICAST={OFF|SEND|RECEIVE|BOTH|ON}]
[POLICYFILTER={100..199|NONE}] [PRIORITYFILTER={200..299|
NONE}] [PROXYARP={ON|OFF}] [RIPMETRIC=1..16] [VJC={ON|
OFF}]
```

The POLICYFILTER parameter specifies the policy filter to be applied. Packets received via the interface are checked against the entries in the policy filter and if a match is found, the packet is routed according to the policy specified in the matching filter entry. Note that a traffic filter, a policy filter and a priority filter can be assigned to an interface. Traffic filters are applied to packets received via the interface, whereas policy and priority filters are applied to packets as they are transmitted. An interface may have a maximum of one traffic filter, one policy filter and one priority filter, but the same traffic, policy or priority filter can be assigned to more than one interface.

The final step is to create static routes and assign policy numbers to the routes, using the ADD IP ROUTE command on page 6-57 or the SET IP ROUTE command on page 6-98:

```
ADD IP ROUTE=ipadd INTERFACE=interface NEXTHOP=ipadd
[CIRCUIT=miox-circuit] [MASK=ipadd] [METRIC=1..16]
[METRIC1=1..16] [METRIC2=1..65535] [POLICY=0..7]
[PREFERENCE=0..65535]
```

```
SET IP ROUTE=ipadd INTERFACE=interface MASK=ipadd
    NEXTHOP=ipadd [CIRCUIT=miox-circuit] [METRIC=1..16]
    [METRIC1=1..16] [METRIC2=1..65535] [POLICY=0..7]
    [PREFERENCE=0..65535]
```

When a packet is received via an interface with an assigned policy filter, and the packet matches an entry in the policy filter, the packet will be routed using a route with the same policy number specified in the matching policy filter entry. For example, if a packet matches a policy filter entry that specifies a POLICY value of 3, the packet will be routed using a route with a POLICY value of 3.

For IP packets routed according to policy numbers 0 to 7, the TOS octet in the packet's IP header is not modified. For IP packets routed according to policy numbers 8 to 15, the TOS field (bits 3 to 6) in the packet's IP header are set to the policy number less 8 and the packet is routed using a route with a policy equivalent to the policy number less 8. For example, if the policy filter assigns an IP packet a policy number of 14, the packet's TOS field is set to 6 (14-8) and the packet is routed using a route with a policy of 6.

Priority-Based Routing

Destination routing protocols such as RIP use metrics to determine the shortest or optimal path to the destination. Priority routing is an alternative mechanism for routing packets according to priorities set by the network manager. Priority routing is used in situations where it is desirable for certain packets to be routed some way other than the obvious shortest path, such as to support low priority batch traffic and high priority interactive traffic over the same link.

To enable priority routing, the network manager defines a set of priorities used to make routing decisions. Each priority specifies the criteria used to select IP packets and the routing actions to perform on IP packets that match the criteria.

The first step is to create a filter to select the IP packets which are to be routed according to priority rather than destination, using a variant of the ADD IP FILTER command on page 6-45:

```
ADD IP FILTER=filter-number SOURCE=ipadd [SMASK=ipadd]
    [SPORT={port}] [DESTINATION=ipadd [DMASK=ipadd]]
    [DPORT={port}] [ICMPCODE={icmp-code}]
    [ICMPTYPE={icmp-type}] [LOG={4..1600|DUMP|HEADER|NONE}]
    [OPTIONS={YES|NO}] [PROTOCOL={protocol|ANY|EGP|ICMP|OSPF|
    TCP|UDP}] [SESSION={ANY|ESTABLISHED|START}] [SIZE=size]
    [ENTRY=entry-number] {ACTION={INCLUDE|EXCLUDE}|
    POLICY=0..15|PRIORITY=P0..P7}
```

IP filters with filter numbers in the range 200 to 299 are treated as priority filters. The ACTION parameter is replaced by the PRIORITY parameter which identifies the priority used to route IP packets which match the filter entry.

The priority filter is then assigned to an interface using the ADD IP INTERFACE command on page 6-53 or the SET IP INTERFACE command on page 6-92:

```
ADD IP INTERFACE=interface IPADDRESS={ipadd|DHCP}
    [BROADCAST={0|1}] [DIRECTEDBROADCAST={YES|NO|ON|OFF}]
    [FILTER={0..99|NONE}] [FRAGMENT={YES|NO}] [MASK=ipadd]
    [METRIC=1..16] [MULTICAST={OFF|SEND|RECEIVE|BOTH|ON}]
    [POLICYFILTER={100..199|NONE}] [PRIORITYFILTER={200..299|
    NONE}] [PROXYARP={ON|OFF}] [RIPMETRIC=1..16] [VJC={ON|
    OFF}]
```



```

SET IP INTERFACE=interface [BROADCAST={0|1}]
    [DIRECTEDBROADCAST={YES|NO|ON|OFF}] [FILTER={0..99|NONE}]
    [FRAGMENT={YES|NO}] [IPADDRESS=ipadd|DHCP] [MASK=ipadd]
    [METRIC=1..16] [MULTICAST={OFF|SEND|RECEIVE|BOTH|ON}]
    [POLICYFILTER={100..199|NONE}] [PRIORITYFILTER={200..299|
    NONE}] [PROXYARP={ON|OFF}] [RIPMETRIC=1..16] [VJC={ON|
    OFF}]

```

The PRIORITYFILTER parameter specifies the priority filter to be applied. Packets received via the interface are checked against the entries in the priority filter and if a match is found, the packet is routed according to the priority specified in the matching filter entry. Note that a traffic filter, a policy filter and a priority filter can be assigned to an interface. Traffic filters are applied to packets received via the interface, whereas policy and priority filters are applied to packets as they are transmitted. An interface may have a maximum of one traffic filter, one policy filter and one priority filter, but the same traffic, policy or priority filter can be assigned to more than one interface.

When a packet is received via the interface, it is checked for a match against the priority filter, and if a match is found the packet is assigned a new priority. When the IP routing module forwards the IP packet the packet is placed in a priority queue determined by the packets assigned priority. Packets in higher priority queues are forwarded ahead of packets in lower priority queues.

Route Templates

IP route templates are used by the router software to add IP routes to IP subnetworks discovered during normal operation by other protocols. This is only required if IP traffic to the discovered IP subnetwork needs to be routed via a route other than the default route.

When a software module other than a routing protocol adds a route to the IP routing table, and is configured to use an IP route template, the software module supplies the IP network address and mask, and the IP route template provides all the other parameters for the entry in the IP route table.

A route template is created or deleted using the commands:

```

ADD IP ROUTE TEMPLATE=name INTERFACE=interface NEXTHOP=ipadd
    [CIRCUIT=miox-circuit] [METRIC=1..16] [METRIC1=1..16]
    [METRIC2=1..65535] [POLICY=0..7] [PREFERENCE=0..65535]

DELETE IP ROUTE TEMPLATE=name

```

An existing route template can be modified using the command:

```

SET IP ROUTE TEMPLATE=name [NEXTHOP=ipadd]
    [CIRCUIT=miox-circuit] [METRIC=1..16] [METRIC1=1..16]
    [METRIC2=1..65535] [POLICY=0..7] [PREFERENCE=0..65535]

```

To display a list of the currently defined route templates, or information about a specific route template, use the command:

```

SHOW IP ROUTE TEMPLATE[=name]

```

Named Hosts

An important function of the IP module is the provision of access to Telnet services. Normally such services are accessed by specifying the IP address of the full domain name of the service provider in the TELNET command on page 7-11 of *Chapter 7, Terminal Server*:

```
TELNET payroll.admin.thecompany.com
TELNET 172.16.8.5
```

A single router may provide access for users to many services. To make access to these services easier for users, the IP module provides a host *nickname* table which maps an IP address or a full domain name to a short, easy to remember nickname. To add entries to the host name table, use the command:

```
ADD IP HOST=name IPADDRESS=ipadd
```

An entry can be modified with the command:

```
SET IP HOST=name IPADDRESS=ipadd
```

or deleted altogether, with the command

```
DELETE IP HOST=name
```

For example, to add the nickname “payroll” for the IP host with IP address 172.16.8.5 and domain name “payroll.thecompany.com”, use the command:

```
ADD IP HOST=payroll IPADDRESS=172.16.8.5
```

If a domain name is specified in the TELNET command on page 7-11 of *Chapter 7, Terminal Server*, when a user tries to access the service, the router will send a Domain Name System (DNS) request to a defined name server to translate the domain name into an IP address. The name server must be defined with the command:

```
SET IP NAMESERVER=ipadd
```

A secondary name server can also be specified with the command:

```
SET IP SECONDARYNAMESERVER=ipadd
```

When the router performs a DNS lookup, it firsts sends the request to the primary name server. If a response is not received within 20 seconds the request is sent to the secondary name server.

Users can now access the service using any of the commands:

```
TELNET payroll
TELNET payroll.admin.thecompany.com
TELNET 172.16.8.5
```

If the *sysName* MIB object is set to the router’s fully qualified domain name (e.g. router.company.com) using the SET SYSTEM NAME command on page 1-55 of *Chapter 1, Operation*, and a name server has been defined using the SET IP NAMESERVER command on page 6-95, then the command:

```
TELNET mainhost
```

will attempt a Telnet connection to the host “mainhost.company.com”, provided “mainhost” is not an IP nickname (IP nicknames take precedence).

DNS Relay Agent

The DNS relay agent receives DNS requests from hosts and forwards them to the router's own configured DNS server. The DNS relay agent is disabled by default, and can be explicitly enabled or disabled using the commands:

```
ENABLE IP DNSRELAY  
DISABLE IP DNSRELAY
```

To display the current state of the DNS relay agent, use the command:

```
SHOW IP
```

Traffic Filters

Filters provide a mechanism for determining whether or not to process IP packets received over network interfaces. When an IP packet matches one of the patterns in a filter, the filter determines whether the packet is discarded or passed to the IP routing module for forwarding.

Filtering is configured on a per-interface basis, for packets received over the interface. Filtering decisions can be based on combinations of source address, destination address, TCP port and protocol.

A *filter* is a list of *patterns*. A *pattern* consists of:

- A half pattern used to compare against the source address and port of an IP packet.
- A half pattern used to compare against the destination address and port of an IP packet.
- A protocol used to compare against the protocol of an IP packet.
- An ICMP message type used to compare against the type field of an ICMP packet.
- A flag used to compare against the presence or absence of an IP options field in an IP packet header.
- A maximum reassembly size used to compare against the reassembled packet size for IP fragments.
- A flag used to compare against the initiating end of a TCP session.
- An action, either *inclusion* or *exclusion*. Inclusion is the action of allowing the IP packet to be processed further and forwarded. Exclusion is the action of discarding the IP packet.

The filter is terminated by an implicit *match all* pattern, with an exclusion action. This pattern can not be removed and does not appear in any displays.

A *half pattern* is a combination of an *IP address*, *network mask* and *port*. The IP address and network mask are represented in dotted decimal notation. The port is a TCP or UDP port number.

A specific half pattern will match exactly one address and port combination. A general half pattern will match a range of addresses and/or ports. When two specific half patterns are combined the resulting specific pattern will match exactly one connection between two specific address/port pairs. Any other

combination of specific and/or general half patterns will produce a general pattern matching more than one address/port pair.

A linear search is performed on the filter. Searching stops at the first match found, so the order of patterns is important. Specific patterns will always appear before general patterns. Within the specific patterns the order of patterns will not affect the filter results, since each pattern matches a specific and exclusive case. Within the general patterns the order of patterns will affect the filter results, since each pattern matches a range of address/port combinations that may overlap with another pattern.

For example, if the aim of the filter is to include all connections from a particular network except for a small range of addresses (e.g. a particular subnet), the exclusion pattern for the subnet must appear before the inclusion pattern for the network. Otherwise, packets from the subnet will be included (and forwarded for processing) by the inclusion pattern without being compared against the exclusion pattern.

Regardless of whether or not a pattern is specific or general, its position in the filter will effect the efficiency of the filter. Patterns matching the most common conditions expected should appear ahead of patterns matching less common conditions, to reduce the number of comparisons required to obtain a match.

An entry is added to a filter with the command:

```
ADD IP FILTER=filter-number SOURCE=ipadd [SMASK=ipadd]
[SPORT={port}] [DESTINATION=ipadd [DMASK=ipadd]]
[DPORT={port}] [ICMPCODE={icmp-code}]
[ICMPTYPE={icmp-type}] [LOG={4..1600|DUMP|HEADER|NONE}]
[OPTIONS={YES|NO}] [PROTOCOL={protocol|ANY|EGP|ICMP|OSPF|
TCP|UDP}] [SESSION={ANY|ESTABLISHED|START}] [SIZE=size]
[ENTRY=entry-number] {ACTION={INCLUDE|EXCLUDE}}
POLICY=0..15|PRIORITY=P0..P7}
```

The SPORT, DPORT, ICMPCODE and ICMPTYPE parameters can be specified as a decimal number or as one of a list of predefined names (see the description of the ADD IP FILTER command on page 6-45). The LOG parameter determines whether or not any matches to a filter entry result in a log message being sent to the router's logging facility, and the content of the log messages.

An entry in a filter can be modified with the command:

```
SET IP FILTER=filter-number ENTRY=entry-number [SOURCE=ipadd]
[SMASK=ipadd] [SPORT={port}] [DESTINATION=ipadd
[DMASK=ipadd]] [DPORT={port}] [ICMPCODE={icmp-code}]
[ICMPTYPE={icmp-type}] [LOG={4..1600|DUMP|HEADER|NONE}]
[OPTIONS={YES|NO}] [PROTOCOL={protocol|ANY|EGP|ICMP|OSPF|
TCP|UDP}] [SESSION={ANY|ESTABLISHED|START}] [SIZE=size]
{ACTION={INCLUDE|EXCLUDE}}|POLICY=0..15|PRIORITY=P0..P7}
```

An entry in a filter can be deleted with the command:

```
DELETE IP FILTER=filter-number ENTRY={entry-number|ALL}
```

The filters and patterns currently defined, and the number of matches, can be displayed with the command:

```
SHOW IP FILTER[=filter-number]
```

For overall efficiency, most traffic received by the router should be forwarded. The router should not be filtering out most of the traffic it receives. The efficiency of the filtering process can be maximised by careful ordering of all

filters, including general filters, to reduce the number of comparisons required for the majority of IP packets. The counts of matches displayed in the output of the `SHOW IP FILTER` command on page 6-117 can aid in determining the most efficient ordering of patterns within filters.

Defining a filter does not automatically enable the filter. To enable a filter the filter must be assigned to a network interface on the router, using the command:

```
ADD IP INTERFACE=interface IPADDRESS={ipadd|DHCP}
[BROADCAST={0|1}] [DIRECTEDBROADCAST={YES|NO|ON|OFF}]
[FILTER={0..99|NONE}] [FRAGMENT={YES|NO}] [MASK=ipadd]
[METRIC=1..16] [MULTICAST={OFF|SEND|RECEIVE|BOTH|ON}]
[POLICYFILTER={100..199|NONE}] [PRIORITYFILTER={200..299|
NONE}] [PROXYARP={ON|OFF}] [RIPMETRIC=1..16] [VJC={ON|
OFF}]
```

To change the filter used on an interface, use the command:

```
SET IP INTERFACE=interface [BROADCAST={0|1}]
[DIRECTEDBROADCAST={YES|NO|ON|OFF}] [FILTER={0..99|NONE}]
[FRAGMENT={YES|NO}] [IPADDRESS=ipadd|DHCP] [MASK=ipadd]
[METRIC=1..16] [MULTICAST={OFF|SEND|RECEIVE|BOTH|ON}]
[POLICYFILTER={100..199|NONE}] [PRIORITYFILTER={200..299|
NONE}] [PROXYARP={ON|OFF}] [RIPMETRIC=1..16] [VJC={ON|
OFF}]
```

The command:

```
SHOW IP INTERFACE
```

displays information about the interfaces assigned to the IP module, including the associated filter (if any) for each interface.

Note that a traffic filter, a policy filter and a priority filter can be assigned to an interface. Traffic filters are applied to packets received via the interface, whereas policy and priority filters are applied to packets as they are transmitted. An interface may have a maximum of one traffic filter, one policy filter and one priority filter, but the same traffic, policy or priority filter can be assigned to more than one interface.

SNMP

SNMP (*Simple Network Management Protocol*) is defined in RFCs 1155–1157, 1213, 1351 and 1352. The router's implementation of SNMP is based on RFC 1157 "*A Simple Network Management Protocol (SNMP)*", and RFC 1812, "*Requirements for IP Version 4 Routers*". SNMP provides a mechanism for management entities, or stations, to extract information from the *Management Information Base (MIB)* of a managed device.

SNMP can use a number of different protocols as its underlying transport mechanism, but the most common transport protocol, and the only one supported by the router, is UDP. Therefore the IP module must be enabled and properly configured in order to use SNMP. SNMP *trap* messages are sent to UDP port 162; all other SNMP messages are sent to UDP port 161. The router's SNMP agent accepts SNMP messages up to the maximum UDP length the router can receive.

The router implements an enterprise MIB (enterprise number 293), and a number of other standard MIBs including MIB-II (RFC 1213), Ethernet-like Interface Types MIB (RFC 1398), and the Host Resources MIB (RFC 1514). See *Chapter 16*,

Simple Network Management Protocol (SNMP) for a detailed description of the router's SNMP agent and the commands required to configure SNMP on the router. See *Appendix C, SNMP MIBs* for a detailed description of the MIBs and objects supported by the router's SNMP agent.



The router's standard SET and SHOW commands can also be used to access objects in the MIBs supported by the router.

Control and Debug Commands

Several commands control the overall operation of the IP module. The IP module is disabled by default. To enable or disable the IP module, use the commands:

```
ENABLE IP
DISABLE IP
```



All setup information is retained if the module is shut down. It is not necessary to enter new setup information after turning on the module.

The IP module operates in one of two modes, SERVER mode or FORWARDING mode. In SERVER mode the router will not route IP packets, but will provide Telnet services, respond to SNMP requests, and use TFTP to download software upgrades. In FORWARDING mode the router will route IP packets, as well as performing all the functions of SERVER mode. The default operational mode is FORWARDING. The operational mode is set with the commands:

```
ENABLE IP FORWARDING
DISABLE IP FORWARDING
```

The current operational mode is retained when the IP module is disabled, and restored when the IP module is re-enabled. A snapshot of the current state of the IP module is displayed with the command:

```
SHOW IP
```

The router stores all setup information (such as IP addresses) in nonvolatile memory. To purge all information stored in the IP module use the command:

```
PURGE IP
```

The commands:

```
ENABLE IP DEBUG
DISABLE IP DEBUG
```

enable and disable the IP debugging facility. When the debugging facility is enabled, any invalid IP packets that are received are stored in a circular buffer for later analysis. The buffer can store up to 40 packets. Subsequent new packets overwrite the oldest existing packets. The buffer can be examined using the command:

```
SHOW IP DEBUG
```

The command:

```
SHOW TCP [=tcb]
```

displays information about the currently active TCP sessions, including the state and port number. If a TCP connection is specified, detailed debugging information for that connection is displayed. The command:

```
SHOW IP UDP
```

displays similar information for the currently active UDP sessions.

Ping and Trace Route

The ping (*Packet Internet Groper*) and trace route functions are used to verify the connections between networks and network devices.

Ping is used to test the connectivity between two network devices to determine whether or not each network device can “see” the other device. The traditional PING command (found on most UNIX systems, for example) can only be used between two systems running the Internet Protocol (IP), and used ICMP *Echo Request* messages. *Echo Request* packets are sent to the destination addresses and responses are recorded. The command:

```
PING [[IPADDRESS=] ipadd] [LENGTH=number] [NUMBER={number |
CONTINUOUS}] [PATTERN=hexnum] [SIPADDRESS=ipadd]
[SCREENOUTPUT={YES|NO}] [TIMEOUT=number] [TOS=number]
```

initiates the transmission of ping packets. Any parameters not specified use the defaults configured with a previous invocation of the command:

```
SET PING [[IPADDRESS=] ipadd] [LENGTH=number] [NUMBER={number |
CONTINUOUS}] [PATTERN=hexnum] [SIPADDRESS=ipadd]
[SCREENOUTPUT={YES|NO}] [TIMEOUT=number] [TOS=number]
```

As each response packet is received a message is displayed on the terminal device from which the command was entered and the details are recorded. The default configuration and summary information can be displayed with the command:

```
SHOW PING
```

The command:

```
STOP PING
```

halts a ping in progress.

Trace route is used to discover the route used to pass packets between two systems running the IP protocol. It sends UDP packets with the Time To Live (TTL) field in the IP header set starting at 1 and increased by one for each subsequent packet sent until the destination is reached. Each hop along the path responds with a TTL exceeded packet and from this the path can be determined. The command:

```
TRACE [[IPADDRESS=] ipadd] [MAXTTL=number] [MINTTL=number]
[NUMBER=number] [PORT=port-number] [SCREENOUTPUT={YES|NO}]
[SOURCE=ipadd] [TIMEOUT=number] [TOS=number]
```

initiates a trace route. Any parameters not specified use the defaults configured with a previous invocation of the command:

```
SET TRACE [[IPADDRESS=] ipadd] [MAXTTL=number] [MINTTL=number]
[NUMBER=number] [PORT=port-number] [SCREENOUTPUT={YES|NO}]
[SOURCE=ipadd] [TIMEOUT=number] [TOS=number]
```

As each response packet is received a message is displayed on the terminal device from which the command was entered and the details are recorded. The default configuration and summary information can be displayed with the command:

```
SHOW TRACE
```

The command:

```
STOP TRACE
```

is used to halt a trace route that is in progress.

Security Options

As well as the security features provided by IP traffic filters (see *"Traffic Filters"* on page 6-21) and restrictions on access to the router's SNMP agent (see *"SNMP"* on page 6-23), the IP module provides a number of features for securing networks.

Source routing of IP packets can be enabled or disabled using the commands:

```
ENABLE IP SRCROUTE  
DISABLE IP SRCROUTE
```

By default, source routing is disabled. Source routing is rarely used for legitimate purposes and is commonly used to circumvent packet-filtering firewalls.

The filtering of IP packets with a small fragment offsets or overlapping fragments can be enabled or disabled using the commands:

```
ENABLE IP FOFILTER  
DISABLE IP FOFILTER
```

Attacks using tiny or overlapping fragments are designed to foil security schemes based only on packet filtering mechanisms. Tiny fragments are too small to contain the full TCP header, making filter pattern matching difficult, while overlapping fragments can be used to 'replace' portions of preceding valid fragments with data that would otherwise be considered 'invalid'.

Broadcast Forwarding

The broadcast forwarding facility provides a mechanism for redirecting UDP broadcast packets to other hosts, routers or networks in an internet. A typical example would be the redirection of NETBIOS broadcasts between a Windows NT server on a central Head Office LAN and Windows NT workstations attached to remote LANs. NETBIOS is only one of a number of UDP broadcast packets that may require forwarding. Others include TFTP, DNS, Time and BOOTP. BOOTP forwarding, defined in RFC 1542, is a special case and is handled separately by the router (see *"BOOTP Relay Agent"* on page 6-29).

Broadcast forwarding is configured by defining, for each interface, a list of one or more UDP ports to listen on, and the destination IP addresses to which any UDP broadcasts are to be forwarded. By default, broadcast forwarding is disabled. When broadcast forwarding is enabled and configured, UDP listen ports are opened for each of the UDP ports on which UDP broadcasts are to be

forwarded. When a UDP broadcast packet is received on an interface for one of the configured ports it will be forwarded to each of the destinations listed for that interface.

Examples

The following examples illustrate two different approaches to configuring broadcast forwarding.

Forwarding to a Unicast Address

In this example, a number of Windows NT workstations on a remote office LAN are attached to a single Windows NT server on the Head Office LAN (Figure 6-3 on page 6-27, Table 6-5 on page 6-27). Since all broadcasts originate from or are intended for a single NT server on the Head Office LAN, the destination address for the NETBIOS port is set to the NT server's unicast IP address. In this case, broadcast forwarding only needs to be configured on the remote router. This method may become administratively difficult if many destinations on the same network must be specified.

Figure 6-3: Example configuration for broadcast forwarding to a unicast address.

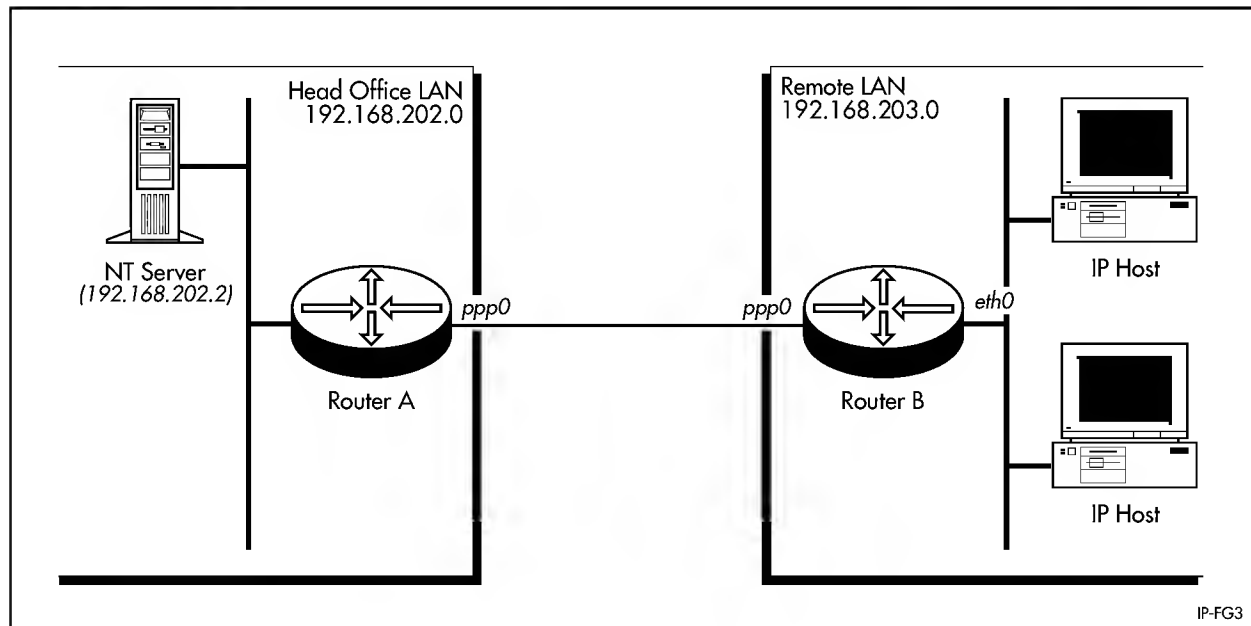
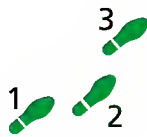


Table 6-5: Example configuration parameters for broadcast forwarding to a unicast address.

Parameter	Head Office	Remote Office
Router Name	A	B
IP address of LAN	192.168.202.0	192.168.203.0
Ethernet interface	eth0	eth0
PPP (WAN link) interface	ppp0	ppp0
IP address of NT Server	192.168.202.2	-
UDP protocol to forward	NETBIOS	-



To configure broadcast forwarding to a unicast address:

1. Enable broadcast forwarding.

```
ENABLE IP
ENABLE IP HELPER
```

2. Configure the UDP protocols to be forwarded.

All NETBIOS broadcasts received by Router B's Ethernet interface eth0 are to be forwarded to the NT server with IP address 192.168.202.2.

```
ADD IP INTERFACE=ETH0 IPADDRESS=192.168.20.1
ADD IP HELPER PORT=NETBIOS DESTINATION=192.168.202.2
INTERFACE=ETH0
```

Forwarding to a Broadcast Address

In this example, a number of Windows NT workstations on a remote office LAN are attached to several Windows NT servers on the Head Office LAN (Figure 6-4 on page 6-28, Table 6-6 on page 6-29). Since broadcasts originate from or are intended for several NT servers on the Head Office LAN, the destination address for the specified port is set to the subnet broadcast address for the Head Office LAN. In this case, broadcast forwarding must be configured on both the remote router and the Head Office router. When the Head Office router receives the UDP packet it re-broadcasts the packet on to the remote LAN. This method has two consequences that need to be considered. Firstly, broadcast traffic will increase on the Head Office LAN. Secondly, if care is not taken with the configuration, broadcast loops may be created. The broadcast forwarding facility in this mode is acting like a pseudo-bridge, but without the protection of protocols such as Spanning Tree to detect loops.

Figure 6-4: Example configuration for broadcast forwarding to a multicast address.

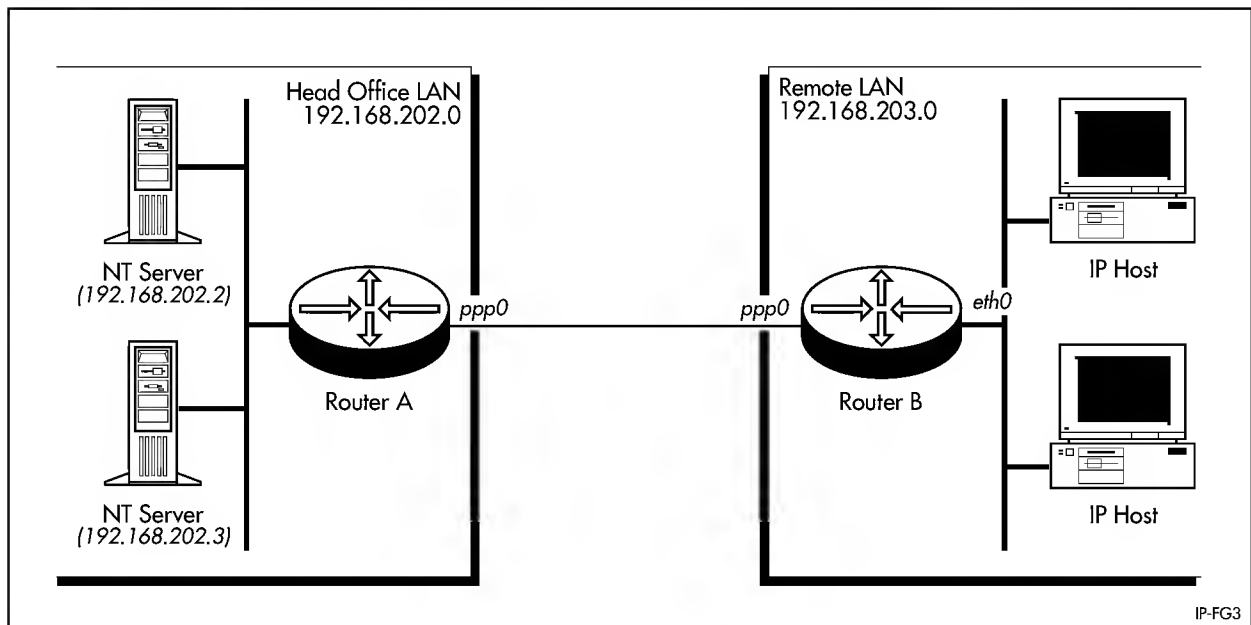
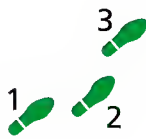


Table 6-6: Example configuration parameters for broadcast forwarding to a multicast address.

Parameter	Head Office	Remote Office
Router Name	A	B
IP address of LAN	192.168.202.0	192.168.203.0
Ethernet interface	eth0	eth0
PPP (WAN link) interface	ppp0	ppp0
IP address of NT Servers	192.168.203.2 192.168.202.3	-
UDP protocol to forward	NETBIOS	-

**To configure broadcast forwarding to a multicast address:****1. Enable broadcast forwarding.**

Enable broadcast forwarding on Router A and Router B, using the following command on each router:

```
ENABLE IP
ENABLE IP HELPER
```

2. Configure the UDP protocols to be forwarded.

All NETBIOS broadcasts received by the remote router's Ethernet interface eth0 are to be forwarded to the Head Office LAN with IP address 192.168.202.0. On Router B, use the command:

```
ADD IP INTERFACE=ETH0 IPADDRESS=192.168.203.2
ADD IP HELPER PORT=NETBIOS DESTINATION=192.168.202.255
INTERFACE=ETH0
```

All NETBIOS broadcasts received by the Head Office router's PPP interface ppp0 are to be broadcast on to the Head Office LAN. On Router A, use the command:

```
CREATE PPP=0 OVER=SYN0
ADD IP INTERFACE=PPP0 IPADDRESS=0.0.0.0
ADD IP HELPER PORT=NETBIOS DESTINATION=192.168.203.255
INTERFACE=PPP0
```

BOOTP Relay Agent

BOOTP is a UDP-based protocol which allows a booting host to configure itself dynamically without external interventions. A BOOTP server responds to requests from BOOTP clients for configuration information, such as the IP address the client should use. BOOTP is defined in RFC 951, "*Bootstrap Protocol (BOOTP)*".

RFC 1542, "*Clarifications and Extensions for the Bootstrap Protocol*", defines extensions to the BOOTP protocol, including the behaviour of a BOOTP Relay Agent.

The router's BOOTP Relay Agent relays BOOTREQUEST messages originating from any of the router's interfaces to a user-defined destination, and relays BOOTREPLY messages addressed to BOOTP clients on networks directly connected to the router. BOOTREPLY messages addressed to clients on

networks not directly connected to the router are ignored by the relay agent and treated as ordinary IP packets for forwarding.

A BOOTREQUEST message may be relayed via unicast, multicast or broadcast methods. In the last case, the message will not be re-broadcast to the interface from which it was received. The relay destinations are configured independently of other broadcast forwarders' destinations (e.g. TFTP).

The 'hops' field in a BOOTP message is used to record the number of hops (routers) the message has been through. If the value of the 'hops' field exceeds a predefined threshold (normally 16), the message will be discarded by the relay agent. The threshold may be set to a value in the range 1 to 16.

The BOOTP Relay Agent is enabled with the command:

```
ENABLE BOOTP RELAY
```

The agent must currently be disabled. The agent can be disabled with the command:

```
DISABLE BOOTP RELAY
```

A relay destination is defined with the command:

```
ADD BOOTP RELAY=ipadd
```

where *ipadd* is the IP address of a BOOTP server in dotted decimal notation. More than one relay destination may be defined, with successive commands. Request messages will be relayed to all defined relay destinations, so messages may be duplicated.

A relay destination may be deleted using the command:

```
DELETE BOOTP RELAY=ipadd
```

where *ipadd* is the IP address of a BOOTP server in dotted decimal notation. The destination must exactly match a destination previously defined with the ADD BOOTP RELAY command on page 6-43.

The BOOTP configuration (including the relay destination list) can be purged with the command:

```
PURGE BOOTP RELAY
```

The BOOTP module is disabled, all configuration data (including nonvolatile storage) is purged, and then BOOTP is re-enabled with default settings.

When the 'hops' field in a BOOTP message exceeds a predefined threshold the BOOTP message is discarded. The default value of the threshold is 4. The threshold may be set to any value in the range 1 to 16 using the command:

```
SET BOOTP MAXHOPS=hops
```

The command:

```
SHOW BOOTP RELAY
```

displays the current configuration of the BOOTP Relay Agent.

IP Multicasting

IP multicasting, defined in RFC 1112 *"Host Extensions for IP Multicasting"* and RFC 1812 *"Requirements for IP version 4 Routers"*, is the process of transmitting an IP datagram to a group of hosts. A *host group* may contain zero or more hosts. A multicast datagram is delivered to each member of the group as if the datagram had been sent individually to each host as a unicast datagram.

A host group is identified by a single IP address. IP addresses in the range 224.0.0.0–239.255.255.255 are reserved for use as multicast addresses, and each address identifies a host group. The IP address 224.0.0.0 is guaranteed not to be assigned to any host group. The IP address 224.0.0.1 is assigned to the permanent group of all IP hosts and gateways, and is used to address all multicast hosts on the directly connected network. There is no multicast IP address for all hosts on the Internet.

Host groups are dynamic—hosts can join or leave host groups at any time. Any host on the Internet can be a member of any host group, and can be a member of any number of groups at the same time. A host does not need to be a member of a host group to send a multicast datagram to the group.

A multicast datagram can be transmitted to both the local network and all remote networks that are reachable within the IP TTL (time-to-live) value for the datagram. To send an IP multicast datagram, a host transmits the datagram as a local multicast datagram to all members of the host group on the directly connected network. Multicast routers on the local network will forward the multicast datagram to all other networks with members in the host group. On the remote destination network the local multicast router will transmit the datagram as a local multicast onto the directly connected network.

The router can be configured to send and receive multicast datagrams, or to only send multicast datagrams, or to only receive multicast, or to neither send nor receive multicast datagrams. IP multicasting can be configured when the IP interface is created, using the command:

```
ADD IP INTERFACE=interface IPADDRESS={ipadd|DHCP}
    [BROADCAST={0|1}] [DIRECTEDBROADCAST={YES|NO|ON|OFF}]
    [FILTER={0..99|NONE}] [FRAGMENT={YES|NO}] [MASK=ipadd]
    [METRIC=1..16] [MULTICAST={OFF|SEND|RECEIVE|BOTH|ON}]
    [POLICYFILTER={100..199|NONE}] [PRIORITYFILTER={200..299|
    NONE}] [PROXYARP={ON|OFF}] [RIPMETRIC=1..16] [VJC={ON|
    OFF}]
```

or configured on an existing IP interface using the command:

```
SET IP INTERFACE=interface [BROADCAST={0|1}]
    [DIRECTEDBROADCAST={YES|NO|ON|OFF}] [FILTER={0..99|NONE}]
    [FRAGMENT={YES|NO}] [IPADDRESS=ipadd|DHCP] [MASK=ipadd]
    [METRIC=1..16] [MULTICAST={OFF|SEND|RECEIVE|BOTH|ON}]
    [POLICYFILTER={100..199|NONE}] [PRIORITYFILTER={200..299|
    NONE}] [PROXYARP={ON|OFF}] [RIPMETRIC=1..16] [VJC={ON|
    OFF}]
```

The state of IP multicasting and counts of multicast packets processed can be displayed using the commands:

```
SHOW IP INTERFACE
SHOW IP INTERFACE COUNTER=MULTICAST
```

Multicast routing is configured on a per-interface basis. All logical IP interfaces on the same IP interface use the same multicast setting, so changing the setting for multicasting on one logical interface will affect all other logical interfaces in the IP interface. For multicast datagrams being forwarded by the router, an IP

interface which has more than one logical interface will only forward one multicast datagram out the interface.

Remote Address Assignment

The remote IP address assignment facility enables unnumbered PPP interfaces (i.e. PPP interfaces with an IP address of 0.0.0.0) to be dynamically assigned an IP address during the PPP link's negotiation process.

If a PPP interface is created with an IP address of 0.0.0.0, and remote IP address assignment is enabled, during the IP control protocol (IPCP) negotiation process the router will allow the remote PPP peer to set the IP address of the local PPP interface.

If the local PPP interface has an IP number other than 0.0.0.0, or if remote IP address assignment is disabled, the router will not allow the remote PPP peer to set the IP address of the local PPP interface.

Remote IP address assignment is enabled or disabled with the commands:

```
ENABLE IP REMOTEASSIGN
DISABLE IP REMOTEASSIGN
```

The current status of the remote IP assignment option is displayed in the output of the SHOW IP command on page 6-106.

IP Address Pools

An IP address pool is a named collection of IP addresses that PPP and other modules can use to assign IP addresses to dynamic connections. The advantage of an address pool is that a finite number of IP addresses can be re-used by many clients. When a client is finished with the IP address the IP address is returned to the pool and is available for another client to use.

The router supports multiple methods for assigning IP addresses to dynamic dial-in calls. The following procedure is used to select the IP address assigned to a dial-in call:

1. If the user is authenticated via the router's internal User Authentication Database, and an IP address is set in the User Authentication Database for that user, then that IP address is used.
2. If the PPP call has an IP pool set, and the request to the IP pool is successful, then that IP address is used.

An IP address pool is created or destroyed using the commands:

```
CREATE IP POOL=pool-name IP=ipadd[-ipadd]
DESTROY IP POOL=pool-name
```

The currently configured IP address pools, and the status of the IP addresses in the pools, can be displayed using the command:

```
SHOW IP POOL[=pool-name] [IP=ipadd[-ipadd]] [SUMMARY]
```

To associate an IP address pool with a PPP interface so that connections using that interface will use IP addresses from the IP address pool, use either of the commands:

```
CREATE PPP=ppp-interface OVER=physical-interface
      IPPOOL=pool-name [other-ppp-options...]

SET PPP=ppp-interface IPPOOL=pool-name [other-ppp-options...]
```

To disassociate an IP address pool from a PPP interface so that connections using that interface will no longer use IP addresses from the IP address pool, use the command:

```
SET PPP=ppp-interface IPPOOL=NONE
```

To associate an IP address pool with a PPP template so that dynamic PPP interfaces created using the PPP template will use IP addresses from the IP address pool, use either of the commands:

```
CREATE PPP TEMPLATE=template IPPOOL=pool-name
      [other-template-options...]

SET PPP TEMPLATE=template IPPOOL=pool-name
      [other-template-options...]
```

To disassociate an IP address pool from a PPP template so that dynamic PPP interfaces created using the PPP template will no longer use IP addresses from the IP address pool, use the command:

```
SET PPP TEMPLATE=template IPPOOL=NONE
```

Configuration Examples

The following examples illustrate the steps required to configure IP on the router. The first example shows how to configure basic IP routing. The second example shows how to configure IP filtering on the router to perform firewall functions.

A Basic TCP/IP Setup

In this example, two routers are to be connected. Each will act as a router rather than just a Telnet server. The routers will be connected to each other using the Point-to-Point Protocol (PPP) over a wide area data communications link. Each router has a single Ethernet LAN segment attached, on which are located local hosts and PCs (Figure 6-5 on page 6-33, Table 6-7 on page 6-34).

Figure 6-5: Example configuration for a basic TCP/IP network.

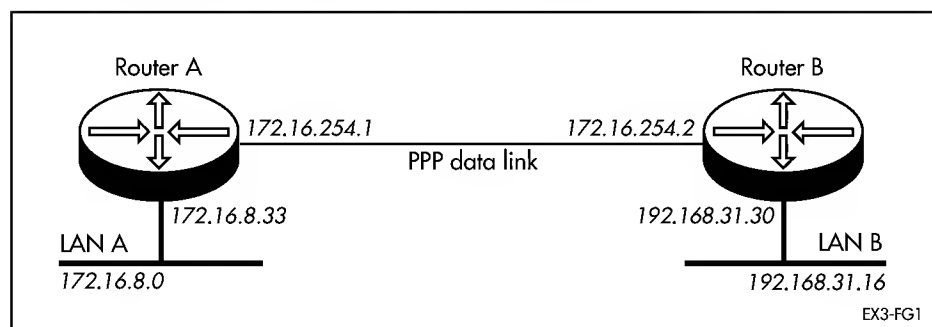
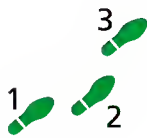


Table 6-7: Example configuration parameters for a basic TCP/IP network.

Parameter	Router A	Router B
LAN IP subnet address	172.16.8.0	192.168.31.16
LAN network class	B	C
LAN number of subnet bits	8	4
LAN IP network mask	255.255.255.0	255.255.255.240
Ethernet IP address	172.16.8.33	192.168.31.30
Synchronous port	0	0
PPP interface	0	0
PPP IP subnet address	172.16.254.0	172.16.254.0
PPP interface IP address	172.16.254.1	172.16.254.2

**To configure a basic IP network:****1. Configure the PPP Link.**

See “The Command Processor” on page 1-3 of *Chapter 1, Operation* for a step-by-step example of how to establish MANAGER level access. See “Configuring a PPP link” on page 3-20 of *Chapter 3, Point-to-Point Protocol (PPP)* for a step-by-step example of how to set up a PPP link.

2. Initialise and enable the IP routing module.

Use the following commands on both routers to initialise the IP routing database and enable the IP routing module. The PURGE IP command disables the IP routing module, so the module must be explicitly enabled:

```
PURGE IP
ENABLE IP
```

3. Add interfaces to the IP routing module.

The interfaces must now be assigned to the IP routing module. Use the following commands on Router A:

```
ADD IP INT=ETH0 IP=172.16.8.33 MASK=255.255.255.0
ADD IP INT=PPP0 IP=172.16.254.1 MASK=255.255.255.0
```

The IP routing module on Router B must now be configured, using a similar sequence of commands. The main difference is that Router B has a Class C network on the Ethernet interface. This requires a different network mask. Use the following commands for Router B:

```
ADD IP INT=ETH0 IP=192.168.31.30 MASK=255.255.255.240
ADD IP INT=PPP0 IP=172.16.254.2 MASK=255.255.255.0
```

The metrics for the interfaces will default to 1. The IP module is now enabled, linked to the physical interfaces and operational. By default, the router will not receive or transmit any route information until it has been configured to use a routing protocol. For this example assume RIP is used.

4. Configure RIP as the routing protocol.

A routing protocol must now be enabled to allow the routers to communicate and to update the internal routing tables. For this example RIP is used. This is to be broadcast onto the Ethernet LAN, but is to be directed explicitly to each end of the PPP link. For Router A, use the following commands:


```
ADD IP RIP INT=ETH0
ADD IP RIP INT=PPP0
SHOW IP RIP
```

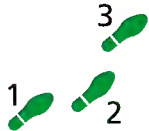


Specifying only the interface causes RIP to be broadcast to the whole network or subnet.

For Router B use:

```
ADD IP RIP INT=ETH0
ADD IP RIP INT=PPP0
SHOW IP RIP
```

The router configuration is now complete.



To test the configuration:

1. Check the operational mode of the IP routing module.

The IP module operates in one of two modes, SERVER or FORWARDING. In SERVER mode the router will not route IP packets, but will provide Telnet services, respond to SNMP requests, and use TFTP to download software upgrades. In FORWARDING mode the router will route IP packets, as well as performing all the functions of SERVER mode. The default operational mode is FORWARDING. To examine the current setting use the command:

```
SHOW IP
```

which displays the general status of the IP module. The operational mode can be changed using the commands:

```
DISABLE IP FORWARDING
ENABLE IP FORWARDING
```



For more information on using the router as a Telnet server see Chapter 7, Terminal Server.

2. Check the routes.

Provided the interfaces are connected to other systems acting as routers the router will obtain IP routes after a short period (up to 60 seconds). These routes show the network from the point of view of the router. The route table can be checked to verify the correct operation of the IP module using the following command on either router:

```
SHOW IP ROUTE
```

This should produce a display (on router A) like that shown in Figure 6-6 on page 6-36.

Figure 6-6: Example output from the SHOW IP ROUTE command for a basic TCP/IP network.

IP Routes					
Destination Circ.	Mask Type	Policy	NextHop Protocol	Interface Metrics	Age Preference
172.16.8.0	255.255.255.0		0.0.0.0	eth0	8372
-	direct	0	static	1	100
172.16.254.0	255.255.255.0		0.0.0.0	ppp0	8372
-	direct	0	static	1	100
192.168.31.16	255.255.255.240		172.16.254.2	ppp0	8369
-	remote	0	rip	2	100

The route table should contain easily verifiable data and should indicate that this router can communicate with other router systems. The PING command on page 6-82 (common to most TCP/IP implementations) can be used on a host to test that paths to remote hosts are available through the router. The router's PING command can be used to verify that hosts respond on both links:

```
PING ipadd
```

or:

```
PING nickname
```

if *nickname* has been added to the host name table using the ADD IP HOST command on page 6-52. ICMP *Echo Request* packets will be sent to the host IP address and the response time for each will be listed if the command is successful.

3. Check the ARP cache.

The ARP cache will start to show binding information (especially from the LAN link) for each active host on the links. The ARP cache can be checked using the command:

```
SHOW IP ARP
```

The router should have entries for some known hosts in the ARP cache. This means that it is able to communicate correctly with these hosts.



Entries will only appear in the ARP cache when a local host attempts to access a host on another subnet, or if it uses a protocol like BOOTP. It is easy to force this to occur by attempting to ping a host on another subnet from a local host.

4. Try using Telnet to access the remote router.

To Telnet from Router A to Router B, on Router A use the command:

```
TELNET 192.168.31.30
```

To Telnet from Router B to Router A, on Router B use the command:

```
TELNET 172.16.8.33
```



You can use any of the assigned interface IP addresses as the target for a Telnet access.

Troubleshooting

No Route Exists to the Remote Router

1. Wait for at least one minute to ensure that a RIP update has been received.
2. Repeat step 4. Check that the link is OPENED for both LCP and IP by typing:

```
SHOW PPP
```

The display should look like Figure 6-7 on page 6-37.

Figure 6-7: Example output from the SHOW PPP command for a basic TCP/IP network.

Name	Enabled	ifIndex	Over	CP	State
ppp0	YES	04		IPCP	OPENED
			isdn-remote	LCP	OPENED

See “Point-to-Point Protocol (PPP)” on page 3-1 of *Chapter 3, Point-to-Point Protocol (PPP)* for further details on how to check the PPP link.

3. Try restarting the IP routing module (a warm restart), by typing:

```
RESET IP
```

If the route still does not appear, contact your distributor or reseller for assistance.

Telnet Fails

1. If Telnet into the remote router fails, check that the IP address being used matches the one assigned to this router. Check that RIP is configured correctly (step 4).
2. If Telnet into a host on the remote LAN fails, but works into the remote router, check the IP address you are using is correct. Check that both routers are gateways, not servers by typing:

```
SHOW IP
```

The “IP Packet Forwarding” entry in the output should be “Enabled”.

3. Ensure that the remote host is running a Telnet daemon and is correctly configured. Check that RIP is being broadcast (i.e. to ‘.255’) on the remote LAN by typing (on the remote router):

```
SHOW IP RIP
```

On Router A the display should look like Figure 6-8 on page 6-37.

Figure 6-8: Example output from the SHOW IP RIP command for a basic TCP/IP network.

Interface	Circuit	IP Address	Send	Receive	Demand	Auth	Password
eth0	–	–	COMP	BOTH	NO	NO	
ppp0	–	172.16.249.34	RIP1	RIP2	YES	NO	
ppp1		172.16.250.2	RIP2	NONE	YES	NO	

4. Check that the ARP cache on the remote router contains an entry for the remote host. This indicates that the host has been active. Use the command:

```
SHOW IP ARP
```

5. Check that a route exists to the subnet that the target host is on, with:

```
SHOW IP ROUTE
```

For sites with multiple subnets on a single LAN, static routes may be required.

6. Try using the PING command on page 6-82 on the remote router to check that the host responds. For example, type:

```
PING 172.16.8.2
```

The response should look like Figure 6-9 on page 6-38.

Figure 6-9: Example output from the PING command.

```
Echo reply 1 from 172.16.8.2 time delay 20 ms
Echo reply 2 from 172.16.8.2 time delay 40 ms
Echo reply 3 from 172.16.8.2 time delay 0 ms
Echo reply 4 from 172.16.8.2 time delay 0 ms
Echo reply 5 from 172.16.8.2 time delay 60 ms
```

7. Contact your distributor or reseller for assistance.

Configuring IP Filters

With the increase in connections to the Internet, and the interconnection of networks from different organisations, filtering of data packets is an important mechanism in ensuring that only legitimate connections are allowed. Security can never be perfect while connections to other networks exist, but filters allow network managers to manage the permissible free access, while restricting users who do not have permission.

The router has firewall functionality and can restrict traffic on the basis of source/destination IP address, source/destination ports, IP protocol type and TCP flags. The exact choice of filters will depend on an organisations particular requirements. However, extensive filtering and large filter lists will reduce the performance of the router, so filtering design needs to ensure that lists are simple, but effective.

In this example, an organisation wishes to allow access to its mainframe for users from another organisation. Access from the remote network will be controlled by filters defined on the local router (Figure 6-10 on page 6-39). On the remote network there are three hosts. Host A can connect via Telnet to the mainframe. Host B can connect via Telnet and FTP. Host C can connect via FTP. Table 6-8 on page 6-39 lists parameter values that will be used in the example. A static route exists for the PPP link between the local and remote routers.

Figure 6-10: Example configuration for IP filtering.

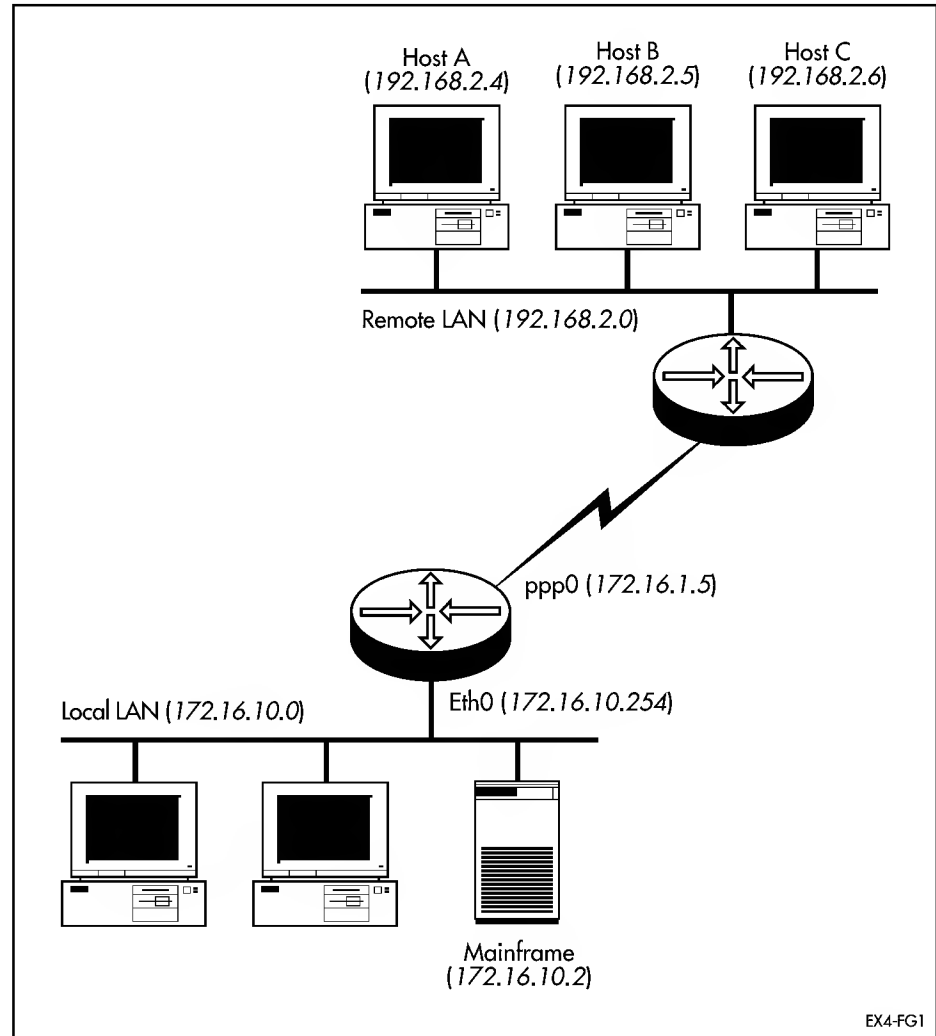
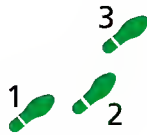


Table 6-8: Example configuration parameters for IP filtering.

Site	Local LAN	Remote LAN
LAN subnet	172.16.10.0	192.168.2.0
LAN network mask	255.255.255.0	255.255.255.0
Eth0 interface IP address	172.16.10.254	-
ppp0 interface IP address	172.16.1.5	-
ppp0 network mask	255.255.255.0	255.255.255.0
Mainframe IP address	172.16.10.2	-
Remote Host A IP address	-	192.168.2.4
Remote Host B IP address	-	192.168.2.5
Remote Host C IP address	-	192.168.2.6



To configure IP filters:

1. Create a filter to control the access of hosts A, B and C to the mainframe.

Create filter 1 for interface ppp0 to control the access of hosts A, B and C on the remote network to the mainframe on the local network. To enable Telnet connections from host A, use the command:

```
ENABLE IP
ADD IP FILT=1 SO=192.168.2.4 SM=255.255.255.255
    DEST=172.16.10.2 DM=255.255.255.255 DPORT=TELNET
    PROT=TCP SESS=ANY AC=INCLUDE
```

To enable Telnet and FTP access from host B, use the commands:

```
ADD IP FILT=1 SO=192.168.2.5 SM=255.255.255.255
    DEST=172.16.10.2 DM=255.255.255.255 DP=FTPDATA
    PROT=TCP SESS=ESTA AC=INCL
ADD IP FILT=1 SO=192.168.2.5 SM=255.255.255.255
    DEST=172.16.10.2 DM=255.255.255.255 DP=FTP PROT=TCP
    SESS=ANY AC=INCL
ADD IP FILT=1 SO=192.168.2.5 SM=255.255.255.255
    DEST=172.16.10.2 DM=255.255.255.255 DP=TELNET PROT=TCP
    SESS=ANY AC=INCL
```

To enable FTP access from host C, use the commands:

```
ADD IP FILT=1 SO=192.168.2.6 SM=255.255.255.255
    DEST=172.16.10.2 DM=255.255.255.255 DP=FTP PROT=TCP
    SESS=ESTA AC=INCL
ADD IP FILT=1 SO=192.168.2.6 SM=255.255.255.255
    DEST=172.16.10.2 DM=255.255.255.255 DP=FTPDATA
    PROT=TCP SESS=ESTA AC=INCL
```

The last entry in a filter is always an implicit entry (one which you do not have to enter) to exclude all sources, destinations and ports. It is equivalent to the command:

```
ADD IP FILT=1 SO=0.0.0.0 SMASK=0.0.0.0 DEST=0.0.0.0.
    DMASK=0.0.0.0 SPORT=ALL ACT=EXCL
```

2. Create a filter to allow only replies from the mainframe to reach hosts A, B and C.

Create filter 2 for interface eth0 to allow the replies from the mainframe to remote hosts A, B and C, but prevent other users on the local network from accessing remote hosts A, B and C:

```
ADD IP FILT=2 SO=172.16.10.2 SM=255.255.255.255 SP=TELNET
    DEST=192.168.2.4 DM=255.255.255.255 PROT=TCP SESS=ESTA
    AC=INCL
ADD IP FILT=2 SO=172.16.10.2 SM=255.255.255.255 SP=TELNET
    DEST=192.168.2.5 DM=255.255.255.255 PROT=TCP SESS=ESTA
    AC=INCL
ADD IP FILT=2 SO=172.16.10.2 SM=255.255.255.255 SP=FTPDATA
    DEST=192.168.2.5 DM=255.255.255.255 PROT=TCP SESS=ANY
    AC=INCL
ADD IP FILT=2 SO=172.16.10.2 SM=255.255.255.255 SP=FTP
    DEST=192.168.2.5 DM=255.255.255.255 PROT=TCP SESS=ESTA
    AC=INCL
ADD IP FILT=2 SO=172.16.10.2 SM=255.255.255.255 SP=FTPDATA
    DEST=192.168.2.65 DM=255.255.255.255 PROT=TCP SESS=ANY
    AC=INCL
ADD IP FILT=2 SO=172.16.10.2 SM=255.255.255.255 SP=FTP
    DEST=192.168.2.6 DM=255.255.255.255 PROT=TCP SESS=ESTA
    AC=INCL
```

The explicit exclusion is not required. Other hosts on the local network will not be able to communicate with hosts on the remote network.

3. Add the filters to the interfaces.

The filters that have been defined must be assigned to interfaces in order for them to take affect. Assign filter 1 to interface ppp0 and filter 2 to interface eth0, using the commands:

```
CREATE PPP=0 OVER=SYN0
ADD IP INT=PPP0 IP=172.16.10.54 MASK=255.255.255.0 FILT=1
ADD IP INT=ETH0 IP=172.16.1.5 MASK=255.255.255.0 FILT=2
```

4. Test the Configuration

The definitions of the filters can be checked with the command:

```
SHOW IP FILTER
```

This will produce a display like Figure 6-11 on page 6-42. The command:

```
SHOW IP INTERFACE
```

displays details of the IP interfaces defined, including the filter assigned to each interface (Figure 6-12 on page 6-42).

Figure 6-11: Example output from the SHOW IP FILTER command for IP filtering.

IP Filters						
No.	Ent.	Source Port Dest. Port Type	Source Address Dest. Address Act/Pol/Pri	Source Mask Dest. Mask Logging	Session Prot. (C/T)	Size Options Matches
1	1	Any 23:23 General	192.168.2.4 172.16.10.2 Include	255.255.255.255 255.255.255.255 Off	Start TCP	Any Any 0
	2	Any 20:20 General	192.168.2.5 172.16.10.2 Include	255.255.255.255 255.255.255.255 Off	Established TCP	Any Any 0
	3	Any 21:21 General	192.168.2.5 172.16.10.2 Include	255.255.255.255 255.255.255.255 Off	Any TCP	Any Any 0
	4	Any 23:23 General	192.168.2.5 172.16.10.2 Include	255.255.255.255 255.255.255.255 Off	Start TCP	Any Any 0
	5	Any 21:21 General	192.168.2.6 172.16.10.2 Include	255.255.255.255 255.255.255.255 Off	Start TCP	Any Any 0
	6	Any 20:20 General	192.168.2.6 172.16.10.2 Include	255.255.255.255 255.255.255.255 Off	Established TCP	Any Any 0
		Requests: 0		Passes: 0		Fails: 0
2	1	23:23 Any General	172.16.10.2 192.168.2.4 Include	255.255.255.255 255.255.255.255 Off	Established TCP	Any Any 0
	2	23:23 Any General	172.16.10.2 192.168.2.5 Include	255.255.255.255 255.255.255.255 Off	Established TCP	Any Any 0
	3	20:20 Any General	172.16.10.2 192.168.2.5 Include	255.255.255.255 255.255.255.255 Off	Any TCP	Any Any 0
	4	21:21 Any General	172.16.10.2 192.168.2.5 Include	255.255.255.255 255.255.255.255 Off	Established TCP	Any Any 0
	5	20:20 Any General	172.16.10.2 192.168.2.65 Include	255.255.255.255 255.255.255.255 Off	Any TCP	Any Any 0
	6	21:21 Any General	172.16.10.2 192.168.2.6 Include	255.255.255.255 255.255.255.255 Off	Established TCP	Any Any 0
		Requests: 0		Passes: 0		Fails: 0

Figure 6-12: Example output from the SHOW IP INTERFACE command for IP filtering.

Interface	Type	IP Address	Bc	Fr	PArp	Filt	RIP	Met.
Pri. Filt	Pol.Filt	Network Mask	MTU	VJC	GRE	DBcast	Mul.	
eth0	Static	172.16.10.254	1	n	On	002	01	
---	---	255.255.255.0	1500	-	---	No	---	
ppp0	Static	172.16.1.5	1	n	-	001	01	
---	---	255.255.255.0	1500	Off	---	No	---	

Command Reference

This section describes the commands available on the router to configure and manage the IP routing module.

See “*Conventions*” on page xxxv of *Preface* in the front of this manual for details of the conventions used to describe command syntax. See *Appendix A, Messages* for a complete list of error messages and their meanings.

ADD BOOTP RELAY

Syntax `ADD BOOTP RELAY=ipadd`

where:

■ *ipadd* is an IP address in dotted decimal notation.

Description This command adds a BOOTP relay destination. The RELAY parameter specifies the IP address of a BOOTP server in dotted decimal notation. Up to 50 relay destinations can be defined, using successive commands. BOOTP request messages are relayed to all defined relay destinations, so messages may be duplicated.

Examples To add the BOOTP server with IP address 192.168.13.11, use:

```
ADD BOOTP RELAY=192.168.13.11
```

See Also DELETE BOOTP RELAY
 DISABLE BOOTP RELAY
 ENABLE BOOTP RELAY
 PURGE BOOTP RELAY
 SET BOOTP MAXHOPS
 SHOW BOOTP RELAY

ADD IP ARP

Syntax `ADD IP ARP=ipadd INTERFACE=interface
{CIRCUIT=miox-circuit | ETHERNET=macadd}`

where:

- *ipadd* is an IP address in dotted decimal notation.
- *interface* is an interface name formed by concatenating a layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number in the range 0 to 15 (e.g. eth0, ppp1-1). If a logical interface is not specified, 0 is assumed (i.e. 'eth0' is equivalent to 'eth0-0').
- *miox-circuit* is the name of a MIOX circuit defined for an X.25 interface, 1 to 15 characters in length. The name is not case-sensitive.
- *macadd* is the physical Ethernet (MAC) address of a host.

Description This command adds a static ARP entry to the ARP cache. This would be used to add entries for hosts which do not support ARP or to speed up the address resolution function for a host. The ARP entry must not already exist.

The ARP parameter specifies the IP address of the host. The INTERFACE parameter specifies the interface over which the host can be reached. The specified interface must already exist. The CIRCUIT and ETHERNET parameters specify the physical address and physical address type for the host. Only one of CIRCUIT or ETHERNET may be specified.

Examples To add an ARP entry for a host with an Ethernet address of 00000008319F and an IP address of 172.16.9.197 on interface eth0, use:

```
ADD IP ARP=172.16.9.197 INT=ETH0 ETHERNET=00000008319F
```

See Also DELETE IP ARP
SET IP ARP
SHOW IP ARP

ADD IP FILTER

Syntax `ADD IP FILTER=filter-number SOURCE=ipadd [SMASK=ipadd]
 [SPORT={port-name|port-id}] [DESTINATION=ipadd
 [DMASK=ipadd]] [DPORT={port-name|port-id}]
 [ICMPCODE={icmp-code-name|icmp-code-id}]
 [ICMPTYPE={icmp-type-name|icmp-type-id}] [LOG={4..1600|
 DUMP|HEADER|NONE}] [OPTIONS={YES|NO}]
 [PROTOCOL={protocol|ANY|EGP|ICMP|OSPF|TCP|UDP}]
 [SESSION={ANY|ESTABLISHED|START}] [SIZE=size]
 [ENTRY=entry-number] {ACTION={INCLUDE|EXCLUDE}|
 POLICY=0..15|PRIORITY=P0..P7}`

where:

- *filter-number* is a number in the range 0 to 299.
- *ipadd* is an IP address in dotted decimal notation.
- *port-name* is the predefined name for an IP port.
- *port-id* is an IP port number, or a range of ports in the form *low:high*.
- *icmp-code-name* is the predefined name for an ICMP reason code.
- *icmp-code-id* is the number of an ICMP reason code.
- *icmp-type-name* is the predefined name of an ICMP message type.
- *icmp-type-id* is the number of an ICMP message type.
- *protocol* is an IP protocol number.
- *size* is a number in the range 64 to 65535.
- *entry-number* is the position of this entry in the filter.

Description This command adds a pattern to an IP traffic filter, policy filter or priority filter. The exact pattern should not already exist in the filter.

The FILTER parameter specifies the number of the filter to which the pattern is to be added. Filters with numbers in the range 0 to 99 are treated as traffic filters, and use the ACTION parameter to specify the action to take with a packet that matches the pattern. Filters with numbers in the range 100 to 199 are treated as policy filters, and use the POLICY parameter to specify the policy to use when routing a packet that matches the pattern. Filters with numbers in the range 200 to 299 are treated as priority filters, and use the PRIORITY parameter to specify the priority to assign to a packet that matches the pattern.

An interface may have a maximum of one traffic filter, one policy filter and one priority filter, but the same traffic, policy or priority filter can be assigned to more than one interface. Traffic filters are applied to packets received via the interface, whereas policy and priority filters are applied to packets as they are transmitted.

The SOURCE parameter specifies the source IP address, in dotted decimal notation, for the pattern.

The SMASK parameter specifies the mask, in dotted decimal notation, to apply to source addresses for this pattern. The mask is used to determine the portion of the source IP address in the IP packet that is significant for comparison with this pattern. The values of SOURCE and SMASK must be compatible. For each bit in SMASK which is set to zero (0) the equivalent bit in SOURCE must also

be zero (0). If either SOURCE or SMASK is 0.0.0.0, both must be 0.0.0.0. The default is 255.255.255.255.

The SPORT parameter specifies the port to check against the source port for this pattern, as the recognised name of a well-known UDP or TCP port (Table 6-9 on page 6-46), a decimal value in the range 0 to 65535, or a range of numbers in the form *low:high*. If *low* is omitted, 0 is assumed. If *high* is omitted, the maximum port number is assumed. If a port other than ANY is specified, the PROTOCOL parameter must also be specified, and must be one of TCP or UDP. The default for SPORT is ANY.

The DESTINATION parameter specifies the destination IP address, in dotted decimal notation, for the pattern. The default is 0.0.0.0.

The DMASK parameter specifies the mask, in dotted decimal notation, to apply to the destination address for this pattern. The mask is used to determine the portion of the destination IP address in the IP packet that is significant for comparison with this pattern. The values of DESTINATION and DMASK must be compatible. For each bit in DMASK which is set to zero (0) the equivalent bit in DESTINATION must also be zero (0). If either DESTINATION or DMASK is 0.0.0.0, both must be 0.0.0.0. If DESTINATION is specified, the default value for DMASK is 255.255.255.255. If DESTINATION is not specified, the default value for DMASK is 0.0.0.0. If DMASK is specified, DESTINATION must also be specified.

The DPORT parameter specifies the port to check against the destination port for this pattern, as the recognised name of a well-known UDP or TCP port (Table 6-9 on page 6-46), a decimal value in the range 0 to 65535, or a range of numbers formatted as *low:high*. If *low* is omitted, 0 is assumed. If *high* is omitted, the maximum port number is assumed. If a port other than ANY is specified, the PROTOCOL parameter must also be specified, and must be one of TCP or UDP. The default for DPORT is ANY.



If a pattern for Telnet is not explicitly added to a filter assigned to an interface, all Telnet traffic received over the specified interface will be discarded. This will prevent Telnet connections to the router itself via the interface. To enable access to the router's command prompt via Telnet, a pattern for Telnet must be added to the filter for the interface.

Table 6-9: Predefined port names used by the IP filtering process.

Port Name	Number	Protocol ¹	Description
ANY	-	-	Any port
BOOTPC	68	UDP	Bootstrap Protocol Client
BOOTPS	67	UDP	Bootstrap Protocol Server
DOMAIN	53	TCP/UDP	Domain Name Server
FINGER	79	TCP	Finger
FTP	21	TCP	File Transfer [Control]
FTPDATA	20	TCP	File Transfer [Default Data]
GOPHER	70	TCP	Gopher
HOSTNAME	101	TCP/UDP	NIC Host Name Server
IPX	213	TCP/UDP	IPX
KERBEROS	88	UDP	Kerberos

Table 6-9: Predefined port names used by the IP filtering process. (Continued)

Port Name	Number	Protocol ¹	Description
LOGIN	49	UDP	Login Host Protocol
MSGICP	29	TCP/UDP	MSG ICP
NAMESERVER	42	UDP	Host Name Server
NEWS	144	TCP	NewS
NNTP	119	TCP	Network News Transfer Protocol
NTP	123	TCP	Network Time Protocol
RTELNET	107	TCP/UDP	Remote Telnet Service
SFTP	115	TCP/UDP	Simple File Transfer Protocol
SMTP	25	TCP	Simple Mail Transfer
SNMP	161	UDP	SNMP
SNMPTRAP	162	UDP	SNMPTRAP
SYSTAT	11	TCP	Active Users
TELNET	23	TCP	Telnet
TFTP	69	UDP	Trivial File Transfer
TIME	37	TCP/UDP	Time
UUCP	540	TCP	uucpd
UUCPRLOGIN	541	TCP/UDP	uucp-rlogin
WWWHTTP	80	TCP	World Wide Web HTTP
XNSTIME	52	TCP/UDP	XNS Time Protocol

¹ The protocol typically used with the port.

The ICMPTYPE and ICMPCODE parameters specify the ICMP message type and ICMP message reason code to match against the ICMP type and code fields in an ICMP packet. The ICMPTYPE parameter specifies the ICMP message type to match, as a decimal value in the range 0 to 65535, or the recognised name of an ICMP type (Table 6-10 on page 6-47). The ICMPCODE parameter specifies the ICMP message reason code to match, as a decimal value in the range 0 to 65535, or the recognised name of an ICMP reason code (Table 6-11 on page 6-48). The ICMPTYPE and ICMPCODE parameters are only valid when the PROTOCOL parameter is set to ICMP.

Table 6-10: Predefined ICMP type names used by the IP filtering process.

ICMP Type Name	ICMP Type Value	ICMP Type Description	ICMP Codes Supported
ECHORPLY	0	Echo reply messages.	No
UNREACHABLE	3	Unreachable messages.	Yes
QUENCH	4	Source quench messages.	No
REDIRECT	5	Redirect messages.	Yes
ECHO	8	Echo request messages.	No
ADVERTISEMENT	9	Router advertisement messages.	No
SOLICITATION	10	Router solicitation messages.	No
TIMEEXCEED	11	Time exceeded messages.	Yes
PARAMETER	12	Parameter problem messages.	Yes

Table 6-10: Predefined ICMP type names used by the IP filtering process.

ICMP Type Name	ICMP Type Value	ICMP Type Description	ICMP Codes Supported
TSTAMP	13	Timestamp request messages.	No
TSTAMPRLY	14	Timestamp reply messages.	No
INFOREQ	15	Information request messages.	No
INFOREP	16	Information reply message.	No
ADDRREQ	17	Address mask request messages.	No
ADDRREP	18	Address mask reply messages.	No
NAMEREQ	37	Name request messages.	No
NAMERPLY	38	Name reply messages.	No

Table 6-11: Predefined ICMP code names used by the IP filtering process.

ICMP Code Name	ICMP Code Value	ICMP Code Description	Applies to ICMP Type Name...
ANY	(any)	Any ICMP code	(any)
NETUNREACH	0	Network unreachable.	UNREACHABLE
HOSTUNREACH	1	Host unreachable.	UNREACHABLE
PROTUNREACH	2	Protocol unreachable.	UNREACHABLE
PORTUNREACH	3	Port unreachable.	UNREACHABLE
FRAGMENT	4	Fragmentation is needed but "do not fragment" flag is set.	UNREACHABLE
SOURCEROUTE	5	Source route failed.	UNREACHABLE
NETUNKNOWN	6	Destination network unknown.	UNREACHABLE
HOSTUNKNOWN	7	Destination host unknown.	UNREACHABLE
HOSTISOLATED	8	Source host isolated	UNREACHABLE
NETCOMM	9	Communication with destination network administratively prohibited.	UNREACHABLE
HOSTCOMM	10	Communication with destination host administratively prohibited.	UNREACHABLE
NETTOS	11	Network unreachable for selected TOS.	UNREACHABLE
HOSTTOS	12	Host unreachable for selected TOS.	UNREACHABLE
FILTER	13	Communication administratively prohibited due to filtering.	UNREACHABLE
HOSTPREC	14	Host precedence violation.	UNREACHABLE
PRECEDENT	15	Precedence cutoff in effect.	UNREACHABLE
NETREDIRECT	0	Redirect datagrams for the network.	REDIRECT
HOSTREDIRECT	1	Redirect datagram for the host.	REDIRECT
NETRTOS	2	Redirect datagrams for the TOS and network.	REDIRECT
HOSTRTOS	3	Redirect datagrams for the TOS and host.	REDIRECT
FRAGREASSM	0	Fragment reassembly time exceeded.	TIMEEXCEED

Table 6-11: Predefined ICMP code names used by the IP filtering process.

ICMP Code Name	ICMP Code Value	ICMP Code Description	Applies to ICMP Type Name...
TTL	1	TTL exceeded in transit.	TIMEEXCEED
PTRPROBLEM	0	Pointer value referencing the octet in the original IP packet caused problem.	PARMETER
NOPTR	1	No pointer present.	PARMETER

The LOG parameter specifies whether or not any matches to a filter entry result in a log message being sent to the router's logging facility, and the content of the log messages. This parameter enables logging of the IP packet filtering process down to the level of an individual filter entry. If a number in the range 4 to 1600 is specified, the filter number, entry number and IP header information (source and destination IP addresses, protocol, source and destination ports, and size) is logged with a message type/subtype of IPFIL/PASS (for patterns with an INCLUDE action) or IPFIL/FAIL (for patterns with an EXCLUDE action). In addition, the first 4 to 1600 octets of the data portion of TCP, UDP and ICMP packets or the first 4 to 1600 octets after the IP header of other protocol packets are logged with a message type/subtype of IPFIL/DUMP. If DUMP is specified, the filter number, entry number and IP header information (source and destination IP addresses, protocol, source and destination ports, and size) is logged with a message type/subtype of IPFIL/PASS (for patterns with an INCLUDE action) or IPFIL/FAIL (for patterns with an EXCLUDE action). In addition, the first 32 octets of the data portion of TCP, UDP and ICMP packets or the first 32 octets after the IP header of other protocol packets are logged with a message type/subtype of IPFIL/DUMP. If HEADER is specified, the filter number, entry number and IP header information (source and destination IP addresses, protocol, source and destination ports, and size) is logged with a message type/subtype of IPFIL/PASS (for patterns with an INCLUDE action) or IPFIL/FAIL (for patterns with an EXCLUDE action). If NONE is specified, matches to the filter entry are not logged. The default is NONE.

The OPTIONS parameter specifies the presence or absence of the IP options field to check against for this pattern. If YES is specified, the pattern matches IP packets with options set. If NO is specified, the pattern matches IP packets without options set. The default is to match IP packets with or without IP options set.

The PROTOCOL parameter specifies the protocol to check against the protocol for this pattern, as a decimal value in the range 0 to 65535, or the recognised name of an IP protocol type (Table 6-12 on page 6-49). If either SPORT or DPORT are specified, PROTOCOL must be defined as TCP or UDP. Specifying TCP or UDP will filter out packets from companion protocols, such as ICMP, RIP and OSPF, that do not use TCP or UDP as a transport mechanism. The default is ANY.

Table 6-12: Predefined protocol names used by the IP filtering process.

Protocol Name	Description
ANY	Any protocol
EGP	Exterior Gateway Protocol
ICMP	Internet Control Message Protocol

Table 6-12: Predefined protocol names used by the IP filtering process. (Continued)

Protocol Name	Description
OSPF	Open Shortest Path First Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol

The SESSION parameter specifies the type of TCP packet to match, and can only be used when the PROTOCOL parameter specifies TCP. If START is specified, the pattern matches TCP packets with the SYN bit set and the ACK bit clear. If ESTABLISHED is specified, the pattern matches TCP packets with either the SYN bit clear or the ACK bit set. If ANY is specified, the pattern matches any TCP packet. The default is ANY.

The SIZE parameter specifies the maximum reassembled size to match against, for each IP fragment. If the fragment's offset plus size is greater than the value specified, the fragment is discarded.

The ENTRY parameter specifies the entry number in the filter which this new pattern occupy. Existing patterns with the same or higher entry numbers will be pushed down the filter. The default is to add the new pattern to the end of the filter.

The ACTION parameter specifies, for traffic filters, the action to take when the pattern is matched. If INCLUSION is specified, the IP packet will be processed and forwarded. If EXCLUSION is specified, the IP packet will be discarded. The ACTION, POLICY and PRIORITY parameters are mutually exclusive—only one may be specified.

The POLICY parameter specifies, for policy filters, the routing policy to use to route packets when the pattern is matched. For policy numbers in the range 0 to 7, only routes with a matching policy will be considered. For policy numbers in the range 8 to 15, only routes with a policy of $n-8$ (where n is the filter policy) will be considered, and the policy value $n-8$ will be written into the TOS field of the packet. The policy number is assigned to incoming packets, but employed during forwarding (transmission). The ACTION, POLICY and PRIORITY parameters are mutually exclusive—only one may be specified.

The PRIORITY parameter specifies, for priority filters, the priority to apply to forwarding packets when the pattern is matched. A low value (P0) assigns a high priority to the packet. A high value (P7) assigns a low priority to the packet. The priority number is assigned to incoming packets, but employed during forwarding (transmission). The ACTION, POLICY and PRIORITY parameters are mutually exclusive—only one may be specified.

Examples To create filters to allow only FTP traffic between two hosts with IP addresses 172.16.10.2 and 192.168.2.6, use the commands:

```
ADD IP FILTER=1 SOURCE=192.168.2.6 SMASK=255.255.255.255
  DESTINATION=172.16.10.2 DPORT=FTP PROTOCOL=TCP
  ACTION=INCLUDE
ADD IP FILTER=1 SOURCE=192.168.2.6 SMASK=255.255.255.255
  DESTINATION=172.16.10.2 DPORT=FTPDATA PROTOCOL=TCP
  ACTION=INCLUDE
ADD IP FILTER=2 SOURCE=172.16.10.2 SMASK=255.255.255.255
  SPORT=FTP DESTINATION=192.168.2.6 PROTOCOL=TCP
  ACTION=INCLUDE
ADD IP FILTER=2 SOURCE=172.16.10.2 SMASK=255.255.255.255
```



```
SPORT=FTPDATA DESTINATION=192.168.2.6 PROTOCOL=TCP  
ACTION=INCLUDE
```

See Also ADD IP ROUTE FILTER
 DELETE IP FILTER
 DELETE IP ROUTE FILTER
 SET IP FILTER
 SHOW IP FILTER
 SHOW IP ROUTE FILTER

ADD IP HELPER

Syntax ADD IP HELPER DESTINATION=*ipadd* INTERFACE=*interface*
 PORT=*port-number*

where:

- *ipadd* is an IP address in dotted decimal notation.
- *interface* is an interface name formed by concatenating a layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number in the range 0 to 15 (e.g. eth0, ppp1-1). If a logical interface is not specified, 0 is assumed (i.e. 'eth0' is equivalent to 'eth0-0').
- *port-number* is a UDP port number in the range 1 to 65535, or one of the predefined UDP port names DNS (port 53), NT or NETBIOS (ports 137 and 138), TACACS (port 49), TIME (port 37) or TFTP (port 69).

Description This command adds a port or a set of named ports to the list of UDP ports to listen for on the specified interface. When a broadcast UDP packet is received on the specified interface with the specified destination port number it is redirected to the destination IP address. This allows all network broadcast packets to be delivered across the internet to appropriate servicing hosts. Multiple invocations of this command can be used for forward packets for several UDP ports to the same IP address, to forward packets for a single UDP port to multiple IP addresses.

The DESTINATION parameter specifies the IP address to which the UDP broadcast traffic will be forward.

The INTERFACE parameter specifies the interface to which the UDP port list is assigned. Only UDP broadcasts received for the specified interface for one of the UDP ports in the UDP port list will be forwarded.

The PORTNUMBER parameter specifies the UDP port, as a decimal number in the range 1 to 65535, or the recognised name of a UDP port set. All broadcast traffic received by the router on the specified port or set of ports will be redirected to the IP host at the destination address. Up to 32 ports can be specified.

Examples To forward all NETBIOS broadcasts received via interface eth0 to IP address 192.168.202.3, use the command:

```
ADD IP HELPER PORT=NETBIOS DESTINATION=192.168.202.3  
INTERFACE=ETH0
```

To forward all broadcasts to UDP port 3001 received via interface eth0 to IP address 192.168.100.2, use the command:

```
ADD IP HELPER PORT=3001 INTERFACE=ETH0  
DESTINATION=192.168.100.2
```

See Also DELETE IP HELPER
DISABLE IP HELPER
ENABLE IP HELPER
SHOW IP HELPER

ADD IP HOST

Syntax ADD IP HOST=*name* IPADDRESS=*ipadd*

where:

- *name* is a character string up to 60 characters in length. If the string contains spaces it must be enclosed in double quotes.
- *ipadd* is an IP address in dotted decimal notation.

Description This command adds a user-defined name for an IP host to the host name table. The host name table makes it easier to Telnet to commonly accessed hosts by enabling the user to enter a shorter, easier to remember name for the host rather than the host's full IP address or domain name. The name can also be used with the PING command on page 6-82.

The HOST parameter specifies the user-defined name for the IP host. A host with the same name must not already exist in the host name table. When a host name is specified in the Telnet command, the entire name will be used to match a name in the host name table. All characters are used in the comparison, including nonalphanumeric characters if they are present.

The IPADDRESS parameter specifies the IP address of the host.

Examples To add the host name "zaphod" to the host name table for an IP host with an IP address of 172.16.1.5 and the domain name "zaphod.company.com", use:

```
ADD IP HOST=Zaphod IP=172.16.1.5
```

To Telnet to the host, use any of the following commands:

```
TELNET zaphod  
TELNET zaphod.company.com  
TELNET 172.16.1.5
```

See Also DELETE IP HOST
SET IP HOST
SET IP NAMESERVER
SET IP SECONDARYNAMESERVER
SHOW IP HOST

ADD IP INTERFACE

Syntax `ADD IP INTERFACE=interface IPADDRESS={ipadd|DHCP}
 [BROADCAST={0|1}] [DIRECTEDBROADCAST={YES|NO|ON|OFF}]
 [FILTER={0..99|NONE}] [FRAGMENT={YES|NO}] [MASK=ipadd]
 [METRIC=1..16] [MULTICAST={OFF|SEND|RECEIVE|BOTH|ON}]
 [POLICYFILTER={100..199|NONE}]
 [PRIORITYFILTER={200..299|NONE}] [PROXYARP={ON|OFF}]
 [RIPMETRIC=1..16] [VJC={ON|OFF}]`

where:

- *interface* is an interface name formed by concatenating a layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number in the range 0 to 15 (e.g. eth0, ppp1-1). If a logical interface is not specified, 0 is assumed (i.e. 'eth0' is equivalent to 'eth0-0').
- *ipadd* is an IP address in dotted decimal notation.

Description This command adds a logical interface to the IP module. The INTERFACE parameter specifies the name of the logical interface, and implicitly, the attached layer 2 interface. The layer 2 interface must already be configured. The IP interface must not already be assigned to the IP module. At least two interfaces must be defined before the router can route IP packets, but only one interface (usually Ethernet) needs to be defined if the router is acting only as a server. A maximum of 512 interfaces can be added. When an interface is added it is automatically enabled. Only one logical interface may be configured to the same IP network or subnet.

The BROADCAST parameter specifies whether or not an all 1's or all 0's broadcast address will be used. The default is 1.



The all 0's setting contravenes current RFC's and is offered for backward compatibility with some older UNIX systems.

The DIRECTEDBROADCAST parameter specifies whether or not the router allows network or subnet broadcasts to be forwarded to the network directly attached to the logical interface. The default is NO.

The FILTER parameter specifies the traffic filter to apply to IP packets transmitted or received over the logical interface. The filter must already have been defined with the ADD IP FILTER command on page 6-45. A logical interface may have a maximum of one traffic filter, one policy filter and one priority filter, but the same traffic, policy or priority filter can be assigned to more than one interface. Traffic filters are applied to packets received via the logical interface. The default is not to apply a filter.

The FRAGMENT parameter specifies whether or not the "Do not fragment" bit will be obeyed for outgoing IP packets that are larger than the MTU of the interface. If YES is specified, the "Do not fragment" bit will be ignored and outgoing IP packets that are larger than the MTU of the interface will be fragmented. This is particularly useful for interfaces configured with encapsulation which can potentially increase packet sizes beyond the MTU of the interface. If NO is specified, the "Do not fragment" bit will be obeyed and IP packets that are larger than the MTU of the interface will be discarded. This is the normal behaviour for IP. The FRAGMENT parameter has no effect on the processing of packets smaller than the interface MTU. The default is NO.

The IPADDRESS parameter specifies the IP address of the logical interface. If DHCP is specified, the router will act as a DHCP client and obtain the configuration of the IP interface via DHCP. Table 6-13 on page 6-54 lists the parameters from the DHCP reply used by the router. If an IP interface is configured to use DHCP to obtain its IP address and subnet mask, the interface will not take part in IP routing until the IP address and subnet mask have been set by DHCP.

Table 6-13: DHCP reply parameters used by the router for configuring IP.

DHCP Parameter	Purpose
IP address and mask	The IP address and subnet mask for the IP interface.
DNS Servers	DNS server addresses are added to the list of IP name servers. A primary name server and a secondary name server are supported. Name servers are normally added manually using the SET IP NAMESERVER and SET IP SECONDARYNAMESERVER commands.
Gateway	A default route is added over the specified interface with the next hop set to the gateway address. If a default route does already exist on the router, the gateway parameter in the DHCP reply is ignored.



Remote address assignment must be enabled using the `ENABLE IP REMOTEASSIGN` command before IP interfaces will accept addresses dynamically assigned by DHCP.

The MASK parameter specifies the subnet mask for the logical interface. The value must be consistent with the value specified for the IPADDRESS parameter. The default is the network mask for the address class of the IP address (e.g. 255.255.0.0 for a Class B address, 255.255.255.0 for a Class C address). If IPADDRESS is set to DHCP, the MASK parameter should not be set as the subnet mask received from the DHCP server will be used.

The MULTICAST parameter specifies how the router will handle multicast packets. If OFF is specified, the router will neither send nor receive multicast packets. If SEND is specified, the router will send but not receive multicast packets. If RECEIVE is specified, the router will receive but not send multicast packets. If BOTH is specified the router will both send and receive multicast packets. The value ON is a synonym for BOTH. Note that this parameter applies to the entire IP interface, not an individual logical interface. Setting the MULTICAST parameter on one logical interface will set the MULTICAST parameter on all other logical interfaces associated with the same IP interface. The default is RECEIVE.

The POLICYFILTER parameter specifies the policy filter to apply to IP packets received over the logical interface. The filter must already have been defined with the ADD IP FILTER command on page 6-45. A logical interface may have a maximum of one traffic filter, one policy filter and one priority filter, but the same traffic, policy or priority filter can be assigned to more than one interface. Policy filters are applied to packets as they are transmitted. The default is not to apply a filter.

The PRIORITYFILTER parameter specifies the priority filter to apply to IP packets received over the logical interface. The filter must already have been defined with the ADD IP FILTER command on page 6-45. A logical interface may have a maximum of one traffic filter, one policy filter and one priority

filter, but the same traffic, policy or priority filter can be assigned to more than one interface. Priority filters are applied to packets as they are transmitted. The default is not to apply a filter.

The PROXYARP parameter enables or disables proxy ARP responses to ARP requests. This parameter is only valid for Ethernet interfaces. The default is ON.

The RIPMETRIC parameter specifies the cost of crossing the logical interface, for RIP. The default is 1. The METRIC parameter is also accepted, for backward compatibility.

The VJC parameter is only valid for Point-to-Point Protocol (PPP) and X25T interfaces, and specifies whether or not Van Jacobson header compression is to be used on the layer 2 interface. The VJC parameter applies to all logical interfaces attached to the same layer 2 interface. Changing the setting on one logical interface will alter the setting on all other logical interfaces attached to the layer 2 interface. Compression provides the most advantage on slower link speeds (up to 48 kbps). At speeds of 64 kbps and higher, compression will actually reduce efficiency and so should be disabled. The default is OFF.



Van Jacobson's TCP/IP header compression should not be enabled on a multilink PPP interface.



For security reasons this command will only be accepted if the user has SECURITY OFFICER privilege.

Examples To add PPP interface 0 (logical interface ppp0-0) with an IP address of 172.16.248.33, a subnet mask of 255.255.255.0, a metric of 5 and Van Jacobson's header compression, use:

```
ADD IP INT=PPP0 IP=172.16.248.33 MASK=255.255.255.0 RIPMET=5
VJC=ON
```

To add a second logical interface to PPP interface 0 (logical interface ppp0-1) with an IP address of 172.16.200.1, a subnet mask of 255.255.255.0, a metric of 5 and Van Jacobson's header compression, use:

```
ADD IP INT=PPP0-1 IP=172.16.200.1 MASK=255.255.255.0 RIPMET=5
VJC=ON
```

See Also DELETE IP INTERFACE
DISABLE IP INTERFACE
ENABLE IP INTERFACE
RESET IP INTERFACE
SET IP INTERFACE
SHOW IP INTERFACE

ADD IP RIP

Syntax `ADD IP RIP INTERFACE=interface [CIRCUIT=miox-circuit]
[IP=ipadd] [SEND={NONE|RIP1|RIP2|COMPATIBLE}]
[RECEIVE={NONE|RIP1|RIP2|BOTH}] [DEMAND={NO|YES}]
[AUTH={NONE|PASSWORD|MD5}] [PASSWORD=password]`

where:

- *interface* is an interface name formed by concatenating a layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number in the range 0 to 15 (e.g. eth0, ppp1-1). If a logical interface is not specified, 0 is assumed (i.e. 'eth0' is equivalent to 'eth0-0').
- *miox-circuit* is the name of a MIOX circuit defined for an X.25 interface, 1 to 15 characters in length. The name is not case-sensitive.
- *ipadd* is an IP address in dotted decimal notation.
- *password* is a character string, 1 to 15 characters in length. It may contain letters (a-z), digits (0-9) and the underscore character ("_").

Description This command adds a RIP neighbour. RIP packets will be sent and/or received to/from an IP address on an interface. One of SEND or RECEIVE must be set to something other than NONE.

The INTERFACE parameter specifies the interface to send or receive RIP packets on.

The CIRCUIT parameter specifies the X.25 circuit on which to send or receive RIP packets. It is a required parameter for X25T interfaces and is valid only when the interface is an X25T interface.

The IP parameter specifies the IP address of the RIP neighbour. If an IP address is specified then RIP packets sent on the interface will be directed to the IP address and RIP packets received on the interface will only be accepted from the address. If no IP address is specified then any packet received on the interface will be accepted and RIP packets sent on the interface will be sent to the IP subnet broadcast address for that interface. If the interface supports broadcasts then the packet will be broadcast.

The SEND parameter specifies the version of RIP packet to send. If NONE is specified then no RIP packets will be sent. If RIP1 is specified then RIP version 1 packets are sent. If RIP2 is specified then RIP version 2 packets are sent. If COMPATIBLE is specified RIP version 2 packets are sent without routes that a router receiving only RIP version 1 will treat as host routes. The default is RIP1.

The RECEIVE parameter specifies the version of RIP packets to receive. If NONE is specified then no RIP packets are accepted from the IP address on the interface. If RIP1 is specified then only RIP version 1 packets are accepted. If RIP2 is specified then only RIP version 2 packets are accepted. If BOTH is specified then either RIP version 1 or RIP version 2 packets are accepted. The default is BOTH.

The DEMAND parameter specifies whether to use the RIP demand procedures when send and receiving RIP and for routes received from this neighbour. If NO is specified the demand procedures are not used. If YES is specified the demand procedures are used. The default is NO.

The AUTHENTICATION parameter specifies the method used to authenticate RIP packets. This must be NONE unless using RIP version 2. If NONE is specified no authentication is used. If PASSWORD is specified then a clear text password is used to authenticate RIP packets. If MD5 is specified then an encrypted password is used to authenticate a RIP packet. The default is NONE.

The PASSWORD parameter specifies the password to use if the AUTHENTICATION parameter is set to PASSWORD or MD5. This parameter is required if authentication is used.

Examples To broadcast RIP version 1 on an Ethernet interface (eth0), use the command:

```
ADD IP RIP INTERFACE=eth0
```

To send RIP version 2 on a demand interface (ppp0) with password authentication, but not accept any RIP packets on the interface, use the command:

```
ADD IP RIP INTERFACE=ppp0 SEND=RIP2 RECEIVE=NONE DEMAND=YES  
AUTHENTICATION=PASSWORD PASSWORD=hanselandgretal
```

To receive RIP version 2 packets on an Ethernet interface (eth0) from one and only one host (172.16.248.33) and broadcast RIP version 1 packets on the interface, use the commands:

```
ADD IP RIP INTERFACE=eth0 IP=172.16.248.33 RECEIVE=RIP2  
SEND=NONE  
ADD IP RIP INTERFACE=eth0 RECEIVE=NONE
```

See Also DELETE IP RIP
SET IP RIP
SHOW IP
SHOW IP RIP

ADD IP ROUTE

Syntax `ADD IP ROUTE=ipadd INTERFACE=interface NEXTHOP=ipadd
[CIRCUIT=miox-circuit] [MASK=ipadd] [METRIC=1..16]
[METRIC1=1..16] [POLICY=0..7] [PREFERENCE=0..65535]`

where:

- *ipadd* is an IP address in dotted decimal notation.
- *interface* is an interface name formed by concatenating a layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number in the range 0 to 15 (e.g. eth0, ppp1-1). If a logical interface is not specified, 0 is assumed (i.e. 'eth0' is equivalent to 'eth0-0').
- *miox-circuit* is the name of a MIOX circuit defined for an X.25 interface, 1 to 15 characters in length. The name is not case-sensitive.

Description This command adds a static route to the IP route table. Static routes can be used to define default routes to external routers or networks. A default route is one with a network address of 0.0.0.0. If the router receives data and can not find a route for it, the data will be sent to the default route. To define a default route IPADDRESS is set to 0.0.0.0 and NEXTHOP points to the network (router) to which default packets are to be directed. The static route must not already

exist. However, if the route exists as a dynamic (e.g. RIP-derived) route, the static route may still be added. A maximum of 300 static routes can be defined.

This command is also used to define subnets. Multiple routes can be defined for a single interface (usually a LAN). This is useful if it is desired to have more than one network or subnet present on a particular interface. A common reason is growth in the number of hosts exceeding the capacity of a single subnet. Additional subnets can be assigned by adding static routes. In this case IPADDRESS is set to the new subnet, NEXTHOP is set to 0.0.0.0, and METRIC should be set to 1.

The ROUTE parameter specifies the IP address of the static route.

The INTERFACE parameter specifies the IP interface with which the route is associated. The interface must already exist and be assigned to the IP module. If the interface is an X.25 DTE interface, the CIRCUIT parameter is required and specifies the name of a MIOX circuit already defined for the X.25 DTE interface.

The NEXTHOP parameter specifies the IP address of the next hop (router) for the route. The default is the IP address of the interface specified by the INTERFACE parameter. For a PPP link, NEXTHOP should be the IP address of the remote end of the PPP link.

The MASK parameter specifies the subnet mask for the route. The default mask is determined using the following algorithm:

1. If MASK is specified, use the specified mask.
2. If the route is the default route, use a mask of 0.0.0.0.
3. If the route is for a network to which the router is not attached, use the unsubnetted mask for the network class (A, B or C).
4. Otherwise, use the subnet mask of the specified interface.

In all cases a check is performed on the route and mask to verify that the route is the same before and after masking. This ensures that a static route is not specified to more than its subnet mask.



In most cases the subnet mask will not need to be specified because the method outlined above will work for most common cases.

The METRIC1 parameter specifies the cost of traversing the route for RIP. The default is 1. The normal range is 2 to 16. A metric of 1 should only be used when adding a subnet to an interface. The METRIC parameter is also accepted, for backward compatibility.

The POLICY parameter specifies the type of service for the route. The default is 0.

The PREFERENCE parameter specifies the preference for the route. When more than one route in the route table matches the destination address in an IP packet, the route with the lowest preference value will be used to route the packet. If two or more candidate routes have the same preference, the route with the longest subnet mask will be used. Interface routes have a preference of 0 and RIP routes have a preference of 100. The default preference for static routes other than 0.0.0.0 is 60. The default for the default static route 0.0.0.0 is 360.

Examples To create a default route that points to a router at the remote end of a PPP link attached to interface ppp0, with the IP address 172.16.8.82, use the command:

```
ADD IP ROUTE=0.0.0.0 INTERFACE=PPP0 NEXTHOP=172.16.8.82
METRIC=1
```

To add the subnet 172.16.9.0 to the existing subnet on interface ETH0:

```
ADD IP ROUTE=172.16.9.0 INTERFACE=ETH0 NEXTHOP=0.0.0.0
METRIC=1
```



Adding static routes to gain more local address space can cause problems with PC based TCP/IP software. The subnet mask on the PC may need to be altered so that the PC can see hosts on other subnets.

See Also DELETE IP ROUTE
SET IP ROUTE
SHOW IP ROUTE

ADD IP ROUTE FILTER

Syntax `ADD IP ROUTE FILTER[=filter-id] IP=ipadd MASK=ipadd
ACTION={INCLUDE|EXCLUDE} [DIRECTION={RECEIVE|SEND|
BOTH}] [INTERFACE=interface] [NEXTHOP=ipadd]
[POLICY=0..7] [PROTOCOL={ANY|RIP|STATIC|INTERFACE}]`

where:

- *filter-id* is a number in the range 1 to 100.
- *ipadd* is an IP address in dotted decimal notation.
- *interface* is an interface name formed by concatenating a layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number in the range 0 to 15 (e.g. eth0, ppp1-1). If a logical interface is not specified, 0 is assumed (i.e. 'eth0' is equivalent to 'eth0-0').

Description This command adds a route filter. A route filter controls which routes are sent and received by the routing protocols.

The FILTER parameter specifies where in the filter list the filter is inserted. If FILTER is not specified the filter is added to the end of the list.

The IP parameter specifies the network address to match. The wildcard character ("*") can be used to match a network range. For example, 192.168.*.* will match all destination networks which start with 192.168. The wildcard character can only be used to replace a complete number—192.168.*.* is valid but 192.16.*.* is not valid.

The MASK parameter specifies the network mask of the network to match. The wildcard character ("*") can be used to match a network mask range. For example, 255.255.*.* will match all destination network masks which start with 255.255. The wildcard character can only be used to replace a complete number—255.255.*.* is valid but 255.25.*.* is not valid.

The ACTION parameter specifies what to do with routes that match the filter. If INCLUDE is specified then the route is included. If EXCLUDE is specified then the route is excluded.

The DIRECTION parameter specifies whether to filter the route when receiving it or when sending it.

The INTERFACE parameter specifies the interface to which the filter applies. If specified the route will only be filtered if the route is sent or received on the interface.

The NEXTHOP parameter specifies the IP address of the next hop router to match. If specified the route will only be filtered if the route is sent or received to or from the next hop.

The POLICY parameter specifies the type of service to filter. If not specified all types of service are filtered.

The PROTOCOL parameter specifies the routing protocol to which the filter applies. If specified the route will only be filtered if the route is sent or received by the specified protocol. The default is ANY.

Examples To add a route filter that includes RIP-derived routes, use the command:

```
ADD IP ROUTE FILT=1 PROT=RIP ACT=INCL DIR=BOTH IP=*. *.*.*
MASK=*. *.*.*.*
```

See Also DELETE IP ROUTE FILTER
SET IP ROUTE FILTER
SHOW IP ROUTE FILTER

ADD IP ROUTE TEMPLATE

Syntax ADD IP ROUTE TEMPLATE=*name* INTERFACE=*interface*
NEXTHOP=*ipadd* [CIRCUIT=*miox-circuit*] [METRIC=1..16]
[METRIC1=1..16] [POLICY=0..7] [PREFERENCE=0..65535]

where:

- *name* is a character string, 1 to 31 characters in length. Valid characters are any printable character. If *name* contains spaces it must be enclosed in double quotes. *name* is not case-sensitive.
- *interface* is an interface name formed by concatenating a layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number in the range 0 to 15 (e.g. eth0, ppp1-1). If a logical interface is not specified, 0 is assumed (i.e. 'eth0' is equivalent to 'eth0-0').
- *ipadd* is an IP address in dotted decimal notation.
- *miox-circuit* is the name of a MIOX circuit defined for an X.25 interface, 1 to 15 characters in length. The name is not case-sensitive.

Description This command adds an IP route template. IP route templates are used by the router software to add IP routes to IP subnetworks discovered during normal operation by other protocols. This is only required if IP traffic to the discovered IP subnetwork needs to be routed via a route other than the default route.

The INTERFACE parameter specifies the IP interface with which any route added using this template is associated. The interface must already exist and be assigned to the IP module. If the interface is an X.25 DTE interface, the CIRCUIT parameter is required and specifies the name of a MIOX circuit already defined for the X.25 DTE interface.

The NEXTHOP parameter specifies the IP address of the next hop (router) for any route added using this template. The default is the IP address of the interface specified by the INTERFACE parameter. For a PPP link, NEXTHOP should be the IP address of the remote end of the PPP link.

The METRIC1 parameter specifies the cost of traversing any route added using this template for RIP. The default is 1. The normal range is 2 to 16. A metric of 1 should only be used when adding a subnet to an interface. The METRIC parameter is also accepted, for backward compatibility.

The POLICY parameter specifies the type of service for any route added using this template. The default is 0.

The PREFERENCE parameter specifies the preference for any route added using this template. When more than one route in the route table matches the destination address in an IP packet, the route with the lowest preference value will be used to route the packet. If two or more candidate routes have the same preference, the route with the longest subnet mask will be used. Interface routes have a preference of 0 and RIP routes have a preference of 100. The default preference for static routes other than 0.0.0.0 is 60. The default for the default static route 0.0.0.0 is 360.

Examples To add an IP route template named "branch_office", use the command:

```
ADD IP ROUTE TEMPLATE=branch_office INTERFACE=PPP0  
NEXTHOP=192.168.23.3
```

See Also DELETE IP ROUTE TEMPLATE
SET IP ROUTE TEMPLATE
SHOW IP ROUTE TEMPLATE

ADD IP TRUSTED

Syntax ADD IP TRUSTED=*ipadd*

where:

■ *ipadd* is an IP address in dotted decimal notation.

Description This command adds an entry to the trusted router table. This table acts as a filter which determines which sources of routing information (RIP) are to be accepted. It would be used in the situation where, for instance, the router is connected to a LAN to which several other routers are connected. It may be desirable for the router to route packets from networks known to the other routers (the usual case). In this case, the other routers broadcast routing information onto the LAN (such as RIP) which is then picked up by the router and used to develop the internal routing table.

In the default case where no trusted routers have been specified, the router will accept all routing information unless the source has been filtered in some way.

For example, it could be filtered using the ADD IP FILTER command on page 6-45. However, this would block all information including routing information from being processed. The related ADD IP ROUTE FILTER command on page 6-59 should be used for filtering only routing information.

The trusted table ensures that the router's routing table is updated only by *trusted* sources of routing information. Other routers will not be filtered, but their routing information will not be used until they are added to the table. A maximum of 32 trusted host addresses can be defined.

The TRUSTED parameter specifies the IP address of a host from which RIP information will be accepted. Adding one or more trusted routers automatically enables the trusted router option. If no trusted routers are defined, the router will accept routing information from any source.

Examples To specify the host with an IP address of 172.16.8.33 as a trusted source of RIP information, use:

```
ADD IP TRUSTED=172.16.8.33
```

See Also ADD IP FILTER
DELETE IP FILTER
DELETE IP TRUSTED
SET IP FILTER
SHOW IP FILTER
SHOW IP TRUSTED

CREATE IP POOL

Syntax CREATE IP POOL=*pool-name* IP=*ipadd*[-*ipadd*]

where:

- *pool-name* is a character string, 1 to 15 characters in length. Valid characters are any printable characters. If *pool-name* contains spaces, it must be enclosed in double quotes.
- *ipadd* is an IP address in dotted decimal notation.

Description This command creates a pool of IP addresses that can be used by PPP and other modules to assign IP addresses.

The POOL parameter specifies a name for the IP address pool. The name is used in other commands to identify the pool.

The IP parameter specifies either a single IP address, or a range of IP addresses to be assigned to the pool. The IP address or IP address range should not overlap with any IP address or address range in any other IP pool.

Examples To create an IP pool named "dialin" with the IP addresses 192.168.1.1 to 192.168.1.16, use the command:

```
CREATE IP POOL=dialin IP=192.168.1.1-192.168.1.16
```

See Also DESTROY IP POOL
SHOW IP POOL

DELETE BOOTP RELAY

Syntax `DELETE BOOTP RELAY=ipadd`

where:

■ *ipadd* is an IP address in dotted decimal notation.

Description This command deletes a BOOTP relay destination. The RELAY parameter specifies the IP address of a BOOTP server in dotted decimal notation.

Examples To delete the BOOTP server with IP address 192.168.13.11, use:

```
DELETE BOOTP RELAY=192.168.13.11
```

See Also ADD BOOTP RELAY
DISABLE BOOTP RELAY
ENABLE BOOTP RELAY
PURGE BOOTP RELAY
SET BOOTP MAXHOPS
SHOW BOOTP RELAY

DELETE IP ARP

Syntax `DELETE IP ARP=ipadd`

where:

■ *ipadd* is an IP address in dotted decimal notation.

Description This command deletes a dynamic or static ARP entry from the ARP cache. The ARP entry must already exist. The ARP parameter specifies the IP address of the ARP entry to be deleted.

Examples To delete an ARP entry for a host with an IP address of 172.16.9.197, use:

```
DELETE IP ARP=172.16.9.197
```

See Also ADD IP ARP
SET IP ARP
SHOW IP ARP

DELETE IP FILTER

Syntax `DELETE IP FILTER=filter-number ENTRY={entry-number|ALL}`

where:

- *filter-number* is a number in the range 0 to 299.
- *entry-number* is the position of this entry in the filter.

Description This command deletes a pattern from an IP traffic filter, policy filter or priority filter. The exact pattern must already exist in the filter.

The FILTER parameter specifies the number of the filter from which the pattern is to be deleted. Filters with numbers in the range 0 to 99 are traffic filters, filters with numbers in the range 100 to 199 are policy filters, and filters with numbers in the range 200 to 299 are priority filters.

The ENTRY parameter specifies the entry number in the filter which is to be deleted. If ALL is specified, all entries in the filter are deleted. Existing patterns with the same or higher entry numbers will be pushed up the filter to occupy the vacant entry.

Examples To delete entry 3 from filter 2, use the command:

```
DELETE IP FILTER=2 ENTRY=3
```

See Also ADD IP FILTER
ADD IP TRUSTED
DELETE IP TRUSTED
SET IP FILTER
SHOW IP FILTER
SHOW IP TRUSTED

DELETE IP HELPER

Syntax `DELETE IP HELPER DESTINATION=ipadd INTERFACE=interface
PORT=port-number`

where:

- *ipadd* is an IP address in dotted decimal notation.
- *interface* is an interface name formed by concatenating a layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number in the range 0 to 15 (e.g. eth0, ppp1-1). If a logical interface is not specified, 0 is assumed (i.e. 'eth0' is equivalent to 'eth0-0').
- *port-number* is a UDP port number in the range 1 to 65535, or one of the predefined UDP port names DNS (port 53), NT or NETBIOS (ports 137 and 138), TACACS (port 49), TIME (port 37) or TFTP (port 69).

Description This command deletes either a port from the list of UDP ports to be forwarded or a destination IP address to which UDP broadcasts are being forwarded.

The DESTINATION parameter specifies the IP address to which the UDP broadcast traffic will be forward.

The INTERFACE parameter specifies the interface to which the UDP port list is assigned. Only UDP broadcasts received for the specified interface for one of the UDP ports in the UDP port list will be forwarded.

The PORTNUMBER parameter specifies the UDP port, as a decimal number in the range 1 to 65535, or the recognised name of a UDP port set. All broadcast traffic received by the router on the specified port or set of ports will be redirected to the IP host at the destination address.

Examples To stop forwarding all NETBIOS broadcasts received via interface eth0 to IP address 192.168.202.3, use the command:

```
DELETE IP HELPER PORT=NETBIOS DESTINATION=192.168.202.3
INTERFACE=ETH0
```

To stop forwarding all broadcasts to UDP port 3001 received via interface eth0 to IP address 192.168.100.2, use the command:

```
DELETE IP HELPER PORT=3001 INTERFACE=ETH0
DESTINATION=192.168.100.2
```

See Also ADD IP HELPER
DISABLE IP HELPER
ENABLE IP HELPER
SHOW IP HELPER

DELETE IP HOST

Syntax DELETE IP HOST=*name*

where:

- *name* is a character string up to 60 characters in length. If the string contains spaces it must be enclosed in double quotes.

Description This command deletes a user-defined name for an IP host from the host name table. The host name table makes it easier to Telnet to commonly accessed hosts by enabling the user to enter a shorter, easier to remember name for the host rather than the host's full IP address or domain name.

The HOST parameter specifies the user-defined name to be deleted. The specified host name must exist in the host name table.

Examples To delete the host name "zaphod" from the host name table, use:

```
DELETE IP HOST=Zaphod
```

See Also ADD IP HOST
SET IP HOST
SET IP NAMESERVER
SET IP SECONDARYNAMESERVER
SHOW IP HOST

DELETE IP INTERFACE

Syntax DELETE IP INTERFACE=*interface*

where:

- *interface* is an interface name formed by concatenating a layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number in the range 0 to 15 (e.g. eth0, ppp1-1). If a logical interface is not specified, 0 is assumed (i.e. 'eth0' is equivalent to 'eth0-0').

Description This command deletes a logical interface from the IP module. The logical interface will no longer be used by the IP routing module.

The INTERFACE parameter specifies the name of the logical interface to be deleted. The interface must already be assigned to the IP routing module. At least two interfaces must be assigned to the IP module for the router to route IP packets, but only one interface (usually Ethernet) needs to be assigned if the router is acting only as a server.



When an IP interface is deleted, any static routes and ARP entries specific to the interface are also deleted.

Examples To delete PPP interface 2, use:

```
DELETE IP INT=PPP2
```

To delete the third logical interface attached to PPP0, use:

```
DELETE IP INT=PPP0-2
```

See Also ADD IP INTERFACE
DISABLE IP INTERFACE
ENABLE IP INTERFACE
RESET IP INTERFACE
SET IP INTERFACE
SHOW IP INTERFACE

DELETE IP RIP

Syntax `DELETE IP RIP INTERFACE=interface [CIRCUIT=miox-circuit]
[IP=ipadd]`

where:

- *interface* is an interface name formed by concatenating a layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number in the range 0 to 15 (e.g. eth0, ppp1-1). If a logical interface is not specified, 0 is assumed (i.e. 'eth0' is equivalent to 'eth0-0').
- *miox-circuit* is the name of a MIOX circuit defined for an X.25 interface, 1 to 15 characters in length. The name is not case-sensitive.
- *ipadd* is an IP address in dotted decimal notation.

Description This command deletes a RIP neighbour. Use this command to stop sending and/or receiving RIP to and/or from a RIP neighbour.

The INTERFACE parameter specifies the interface via which RIP packets are received from the RIP neighbour.

The CIRCUIT parameter specifies the X.25 circuit on which to send or receive RIP packets. It is a required parameter for X25T interfaces and is valid only when the interface is an X25T interface.

The IP parameter specifies the IP address of the neighbour to delete.

Examples To delete a neighbour that is broadcasting RIP on an Ethernet interface, use the command:

```
DELETE IP RIP INTERFACE=eth0
```

To delete a neighbour that is sending to a specific IP address on a PPP interface, use the command:

```
DELETE IP RIP INTERFACE=ppp0 IP=172.16.248.33
```

See Also ADD IP RIP
SET IP RIP
SHOW IP
SHOW IP RIP

DELETE IP ROUTE

Syntax `DELETE IP ROUTE=ipadd MASK=ipadd INTERFACE=interface
NEXTHOP=ipadd`

where:

- *ipadd* is an IP address in dotted decimal notation.
- *interface* is an interface name formed by concatenating a layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number in the range 0 to 15 (e.g. eth0, ppp1-1). If a logical interface is not specified, 0 is assumed (i.e. 'eth0' is equivalent to 'eth0-0').

Description This command deletes a static route from the IP route table. The static route must already exist. However, if the route exists as a dynamic (e.g. RIP-derived) route, the static route may not be deleted. A maximum of 300 static routes can be defined.

The ROUTE parameter specifies the IP address of the static route.

The INTERFACE parameter specifies the IP interface with which the route is associated. The interface must already exist and be assigned to the IP module. If the interface is an X.25 DTE interface, the CIRCUIT parameter is required and specifies the name of a MIOX circuit already defined for the X.25 DTE interface.

The NEXTHOP parameter specifies the IP address of the next hop (router) for the route. The default is the IP address of the interface specified by the INTERFACE parameter. For a PPP link, NEXTHOP should be the IP address of the remote end of the PPP link.

The MASK parameter specifies the subnet mask for the route. A check is performed on the route and mask to verify that the route is the same before and after masking. This ensures that a static route is not specified to more than its subnet mask.

Examples To delete a default route that points to a router at the remote end of a PPP link attached to interface ppp0, with the IP address 172.16.8.82, use the command:

```
DELETE IP ROUTE=0.0.0.0 MASK=0.0.0.0 INTERFACE=PPP0  
NEXTHOP=172.16.8.82
```

See Also ADD IP ROUTE
SET IP ROUTE
SHOW IP ROUTE

DELETE IP ROUTE FILTER

Syntax `DELETE IP ROUTE FILTER=filter-id`

where:

- *filter-id* is a number in the range 1 to 100.

Description This command deletes a route filter. A route filter controls which routes are sent and received by the routing protocols.

The FILTER parameter specifies the index in the filter list of the filter to delete. The specified entry must exist.

Examples To delete router filter 3, use the command:

```
DELETE ROUTE FILTER=3
```

See Also ADD IP ROUTE FILTER
SET IP ROUTE FILTER
SHOW IP ROUTE FILTER

DELETE IP ROUTE TEMPLATE

Syntax `DELETE IP ROUTE TEMPLATE=name`

where:

- *name* is a character string, 1 to 31 characters in length. Valid characters are any printable character. If *name* contains spaces it must be enclosed in double quotes. *name* is not case-sensitive.

Description This command deletes the specified IP route template.

Examples To delete an IP route template named "branch_office", use the command:

```
DELETE IP ROUTE TEMPLATE=branch_office
```

See Also ADD IP ROUTE TEMPLATE
SET IP ROUTE TEMPLATE
SHOW IP ROUTE TEMPLATE

DELETE IP TRUSTED

Syntax `DELETE IP TRUSTED=ipadd`

where:

- *ipadd* is an IP address in dotted decimal notation.

Description This command deletes an entry from the trusted router table. This table acts as a filter which determines which sources of routing information (RIP) are to be

accepted. It would be used in the situation where, for instance, the router is connected to a LAN to which several other routers are connected. It may be desirable for the router to route packets from networks known to the other routers (the usual case). In which case, the other routers broadcast routing information onto the LAN (such as RIP) which is then picked up by the router and used to develop the internal routing table.

In the default case where no trusted routers have been specified, the router will accept all routing information unless the source has been filtered in some way. For example, it could be filtered using the ADD IP FILTER command on page 6-45. However, this would block all information including routing information from being processed. The related ADD IP ROUTE FILTER command on page 6-59 should be used for filtering only routing information.

The trusted table ensures that the router's routing table is updated only by *trusted* sources of routing information. Other routers will not be filtered, but their routing information will not be used until they are added to the table.

The TRUSTED parameter specifies the IP address of a host from which RIP information will no longer be accepted. Deleting all trusted routers automatically disables the trusted router option.

Examples To delete the host with an IP address of 172.16.8.33 as a trusted source of RIP information, use:

```
DELETE IP TRUSTED=172.16.8.33
```

See Also ADD IP FILTER
ADD IP TRUSTED
DELETE IP FILTER
SET IP FILTER
SHOW IP FILTER
SHOW IP TRUSTED

DELETE TCP

Syntax DELETE TCP=*tcb*

where:

■ *tcb* is the index of a TCP connection in the TCP connection table.

Description This command deletes an active TCP session. The TCP parameter specifies the index in the TCP connection table of the TCP connection to be deleted. The index can be obtained from the output of the SHOW TCP command on page 6-139. TCP sessions in the LISTEN state can not be deleted.

Examples To delete TCP session number 6, use the command:

```
DELETE TCP=6
```

See Also SHOW TCP

DESTROY IP POOL

Syntax DESTROY IP POOL=*pool-name*

where:

- *pool-name* is a character string, 1 to 15 characters in length. Valid characters are any printable characters. If *pool-name* contains spaces, it must be enclosed in double quotes.

Description This command destroys an existing IP address pool. An IP address pool can only be destroyed when there are no IP addresses in use.

The POOL parameter specifies the name of the IP address pool. The specified address pool must already exist.

Examples To destroy the IP pool named "dialin", use the command:

```
DESTROY IP POOL=dialin
```

See Also CREATE IP POOL
SHOW IP POOL

DISABLE BOOTP RELAY

Syntax DISABLE BOOTP RELAY

Description This command disables the BOOTP Relay Agent. The BOOTP Relay Agent relays BOOTREQUEST messages originating from any of the router's interfaces to a user-defined destination, and relays BOOTREPLY messages addressed to BOOTP clients on networks directly connected to the router. BOOTREPLY messages addressed to clients on networks not directly connected to the router are ignored by the relay agent and treated as ordinary IP packets for forwarding. The BOOTP Relay Agent is disabled by default.

Examples To disable the BOOTP relay agent, use the command:

```
DISABLE BOOTP RELAY
```

See Also ADD BOOTP RELAY
DELETE BOOTP RELAY
ENABLE BOOTP RELAY
PURGE BOOTP RELAY
SET BOOTP MAXHOPS
SHOW BOOTP RELAY

DISABLE IP

Syntax `DISABLE IP`

Description This command disables the IP routing module. The IP module must currently be disabled. The router will no longer route IP packets, respond to SNMP requests, use TFTP to download software upgrades, or provide Telnet services. By default the IP module is disabled.

The IP module operates in one of two modes, SERVER mode or FORWARDING mode. In SERVER mode the router will not route IP packets, but will provide Telnet services, respond to SNMP requests, and use TFTP to download software upgrades. In FORWARDING mode the router will route IP packets, as well as performing all the functions of SERVER mode. The default operational mode is FORWARDING.

The current operational mode of the IP module, SERVER or FORWARDING, is retained and will be restored when the IP module is next enabled.

See Also `DISABLE IP FORWARDING`
 `DISABLE IP SRCROUTE`
 `ENABLE IP`
 `ENABLE IP FORWARDING`
 `ENABLE IP SRCROUTE`
 `SHOW IP`

DISABLE IP DEBUG

Syntax `DISABLE IP DEBUG`

Description This command disables the IP debugging facility. Incorrectly formatted IP packets are buffered for later diagnosis. Up to 40 packets can be stored in the buffer, with subsequent packets replacing the oldest packets. The packet headers can be displayed with the `SHOW IP DEBUG` command on page 6-116. The IP debugging facility must currently be enabled. The debugging facility is disabled by default.

See Also `ENABLE IP DEBUG`
 `SHOW IP DEBUG`
 `SHOW IP`

DISABLE IP DNSRELAY

Syntax `DISABLE IP DNSRELAY`

Description This command disables the DNS relay agent. The router will stop forwarding DNS requests from hosts to the router's own configured DNS server. The DNS relay agent is disabled by default.

See Also `ENABLE IP DNSRELAY`
`SHOW IP`

DISABLE IP ECHOREPLY

Syntax `DISABLE IP ECHOREPLY`

Description This command disables the generation of ICMP *Echo Reply* messages in response to ICMP *Echo Request* messages. The generation of ICMP *Echo Reply* messages is enabled by default.

See Also `ENABLE IP ECHOREPLY`

DISABLE IP FOFILTER

Syntax `DISABLE IP FOFILTER`

Description This command disables the filtering (discarding) of IP packets with a fragment offset of 1, and is intended for use in secure environments to prevent attacks by intruders using *tiny fragments* or *overlapping fragments* (see RFC 1858 for a detailed description). By default, the filter is enabled.

In the *tiny fragment* attack, the attacker transmits IP packets of the minimum fragment size. The first packet contains the IP header and only 8 octets of data, which is insufficient to hold a complete TCP header. The TCP flags field is forced into the second fragment. Filters that attempt to discard connection requests (TCP datagrams with the SYN bit set and the ACK bit clear) will be unable to test these flags in the first fragment and will typically ignore them in subsequent fragments. As a result the IP packet is not discarded. The fragment offset filter discards all fragments with a fragment offset of one, preventing reassembly of the packet at the receiving host.

In the *overlapping fragment* attack, the attacker transmits IP packets in fragments which overlap, again in an attempt to circumvent filters which discard connection requests. The first fragment contains a complete TCP header (so it avoids filters which discard fragments with a fragment offset of one) with the SYN bit clear and the ACK bit set (so it passes filters which discard connection requests). The second fragment has an offset of eight octets and contains another set of TCP flags, this time with the SYN bit set and the ACK bit clear. Typically, this fragment is passed by the filter, and at the receiving host the reassembly process results in the second fragment partially overwriting the first fragment.

The fragment offset filter discards all fragments with a fragment offset of one, preventing reassembly of the packet at the receiving host and effectively preventing both tiny fragment and overlapping fragment attacks.



If IP traffic filters have been created to drop connection requests (using the `SESSION=START` parameter of the `ADD IP FILTER` command on page 6-45 or the `SET IP FILTER` command on page 6-88), the fragment offset filter should be enabled to prevent tiny fragment and offset fragment attacks from circumventing the IP traffic filters.

See Also `ADD IP FILTER`
 `DELETE IP FILTER`
 `ENABLE IP FOFILTER`
 `SET IP FILTER`
 `SHOW IP FILTER`

DISABLE IP FORWARDING

Syntax `DISABLE IP FORWARDING`

Description This command sets the IP module's operational mode to `SERVER`, disabling the forwarding (routing) function. The IP module must currently be enabled and in `FORWARDING` mode.

The IP module operates in one of two modes, `SERVER` mode or `FORWARDING` mode. In `SERVER` mode the router will not route IP packets, but will provide Telnet services, respond to SNMP requests, and use TFTP to download software upgrades. In `FORWARDING` mode the router will route IP packets, as well as performing all the functions of `SERVER` mode. The default operational mode is `FORWARDING`.

See Also `DISABLE IP`
 `DISABLE IP SRCROUTE`
 `ENABLE IP`
 `ENABLE IP FORWARDING`
 `ENABLE IP SRCROUTE`
 `SHOW IP`

DISABLE IP HELPER

Syntax `DISABLE IP HELPER`

Description This command disables the forwarding of broadcast UDP traffic on specified UDP ports to specified destination IP addresses.

Examples To disable broadcast forwarding, use the command:

```
DISABLE IP HELPER
```


See Also ADD IP HELPER
 DELETE IP HELPER
 ENABLE IP HELPER
 SHOW IP HELPER

DISABLE IP INTERFACE

Syntax DISABLE IP INTERFACE=*interface*

where:

- *interface* is an interface name formed by concatenating a layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number in the range 0 to 15 (e.g. eth0, ppp1-1). If a logical interface is not specified, 0 is assumed (i.e. 'eth0' is equivalent to 'eth0-0').

Description This command temporarily disables a logical IP interface. The logical interface will not be used by the IP routing module. The effect is equivalent to physically disconnecting the router from the attached IP network. Routes associated with a disabled interface are not explicitly deleted. However, routes learned via a routing protocol such as RIP will eventually be deleted by the routing protocol's aging mechanism. Static routes are retained until explicitly removed by deleting the specific static route entry or by deleting the IP interface.

The INTERFACE parameter specifies the name of the logical interface to be disabled. The interface must be assigned to the IP routing module and currently enabled.

Examples To disable the first logical IP interface attached to PPP0, use the command:

```
DISABLE IP INT=PPP0-0
```

See Also ADD IP INTERFACE
 DELETE IP INTERFACE
 ENABLE IP INTERFACE
 RESET IP INTERFACE
 SET IP INTERFACE
 SHOW IP INTERFACE

DISABLE IP REMOTEASSIGN

Syntax DISABLE IP REMOTEASSIGN

Description This command disables the remote assignment of IP addresses for unnumbered PPP interfaces. If a PPP interface is created with an IP address of 0.0.0.0, and remote IP address assignment is enabled, during the IP control protocol (IPCP) negotiation process the router will allow the remote PPP peer to set the IP address of the local PPP interface. If the local PPP interface has an IP number other than 0.0.0.0, or if remote IP address assignment is disabled, the router will not allow the remote PPP peer to set the IP address of the local PPP interface.

Examples To disable remote IP address assignment, use the command:

```
DISABLE IP REMOTEASSIGN
```

See Also ENABLE IP REMOTEASSIGN
SHOW IP

DISABLE IP ROUTE

Syntax ENABLE IP ROUTE {CACHE|COUNT|MULTIPATH}

Description This command disables route caching, route counters or equal cost multipath routing.

The CACHE parameter disables route caching. The cache is enabled by default.

The COUNT parameter disables counting of octets sent and received to and from a network. It is disabled by default.

The MULTIPATH parameter disables equal cost multipath routing whereby if several routes with the same cost, policy and type exist for a destination, all traffic for that destination will be equally shared among the routes.

Examples To disable equal cost multipath routing, use the command:

```
DISABLE IP ROUTE MULTIPATH
```

See Also ENABLE IP ROUTE
SHOW IP ROUTE

DISABLE IP SRCROUTE

Syntax DISABLE IP SRCROUTE

Description This command disables the forwarding of source-routed IP packets.

When forwarding is enabled, source-routed IP packets will be forwarded by the router as normal. When forwarding disabled, source-routed packets will be discarded by the router. The default is to disable forwarding, and source-routed IP packets will be discarded.

Source routing is rarely used for legitimate purposes, and is a common method used to circumvent packet-filtering firewalls and to masquerade as a trusted host inside the destination network. This command is therefore an extra security feature, which is why source routed packets are discarded by default.

When the forwarding of IP source routed datagrams is disabled, all source routed packets are logged to the logging facility with a message type/subtype of IPFIL/SRCRT.

See Also DISABLE IP
ENABLE IP
ENABLE IP FORWARDING
ENABLE IP SRCROUTE
SHOW IP

ENABLE BOOTP RELAY

Syntax ENABLE BOOTP RELAY

Description This command enables the BOOTP Relay Agent. The BOOTP Relay Agent relays BOOTREQUEST messages originating from any of the router's interfaces to a user-defined destination, and relays BOOTREPLY messages addressed to BOOTP clients on networks directly connected to the router. BOOTREPLY messages addressed to clients on networks not directly connected to the router are ignored by the relay agent and treated as ordinary IP packets for forwarding. The BOOTP Relay Agent is disabled by default.

See Also ADD BOOTP RELAY
DELETE BOOTP RELAY
DISABLE BOOTP RELAY
PURGE BOOTP RELAY
SET BOOTP MAXHOPS
SHOW BOOTP RELAY

ENABLE IP

Syntax ENABLE IP

Description This command enables the IP routing module. The IP module must currently be disabled. The IP module is disabled by default.

The operational mode of the IP module, SERVER or FORWARDING, is restored to the mode when the IP module was last disabled. The default mode is FORWARDING.

The IP module operates in one of two modes, SERVER mode or FORWARDING mode. In SERVER mode the router will not route IP packets, but will provide Telnet services, respond to SNMP requests, and use TFTP to download software upgrades. In FORWARDING mode the router will route IP packets, as well as performing all the functions of SERVER mode. The default operational mode is FORWARDING.

See Also DISABLE IP
DISABLE IP FORWARDING
DISABLE IP SRCROUTE
ENABLE IP FORWARDING
ENABLE IP SRCROUTE
SHOW IP

ENABLE IP DEBUG

Syntax `ENABLE IP DEBUG`

Description This command enables the IP debugging facility. Incorrectly formatted IP packets are buffered for later diagnosis. Up to 40 packets can be stored in the buffer, with subsequent packets replacing the oldest packets. The packet headers can be displayed with the `SHOW IP DEBUG` command on page 6-116. The IP debug facility must currently be disabled. The IP debugging facility is disabled by default.

See Also `DISABLE IP DEBUG`
`SHOW IP DEBUG`
`SHOW IP`

ENABLE IP DNSRELAY

Syntax `ENABLE IP DNSRELAY`

Description This command enables the DNS relay agent. The router will forward DNS requests from hosts to the router's own configured DNS server. The DNS relay agent is disabled by default.

See Also `DISABLE IP DNSRELAY`
`SHOW IP`

ENABLE IP ECHOREPLY

Syntax `ENABLE IP ECHOREPLY`

Description This command enables the generation of ICMP *Echo Reply* messages in response to ICMP *Echo Request* messages. The generation of ICMP *Echo Reply* messages is enabled by default.

See Also `DISABLE IP ECHOREPLY`

ENABLE IP FOFILTER

Syntax `ENABLE IP FOFILTER`

Description This command enables the filtering (discarding) of IP packets with a fragment offset of 1, and is intended for use in secure environments to prevent attacks by intruders using *tiny fragments* or *overlapping fragments* (see RFC 1858 for a detailed description). By default, the filter is enabled.

In the *tiny fragment* attack, the attacker transmits IP packets of the minimum fragment size. The first packet contains the IP header and only 8 octets of data, which is insufficient to hold a complete TCP header. The TCP flags field is forced into the second fragment. Filters that attempt to discard connection requests (TCP datagrams with the SYN bit set and the ACK bit clear) will be unable to test these flags in the first fragment and will typically ignore them in subsequent fragments. As a result the IP packet is not discarded. The fragment offset filter discards all fragments with a fragment offset of one, preventing reassembly of the packet at the receiving host.

In the *overlapping fragment* attack, the attacker transmits IP packets in fragments which overlap, again in an attempt to circumvent filters which discard connection requests. The first fragment contains a complete TCP header (so it avoids filters which discard fragments with a fragment offset of one) with the SYN bit clear and the ACK bit set (so it passes filters which discard connection requests). The second fragment has an offset of eight octets and contains another set of TCP flags, this time with the SYN bit set and the ACK bit clear. Typically, this fragment is passed by the filter, and at the receiving host the reassembly process results in the second fragment partially overwriting the first fragment.

The fragment offset filter discards all fragments with a fragment offset of one, preventing reassembly of the packet at the receiving host and effectively preventing both tiny fragment and overlapping fragment attacks.

IP datagrams discarded by the fragment offset filter are logged to the logging facility with a message type/subtype of IPFIL/FRAG.



If IP traffic filters have been created to drop connection requests (using the SESSION=START parameter of the ADD IP FILTER command on page 6-45 or the SET IP FILTER command on page 6-88), the fragment offset filter should be enabled to prevent tiny fragment and offset fragment attacks from circumventing the IP traffic filters.

See Also ADD IP FILTER
 DELETE IP FILTER
 DISABLE IP FOFILTER
 SET IP FILTER
 SHOW IP FILTER

ENABLE IP FORWARDING

Syntax ENABLE IP FORWARDING

Description This command sets the IP module's operational mode to FORWARDING, enabling the forwarding (routing) function. The IP module must currently be enabled and in SERVER mode.

The IP module operates in one of two modes, SERVER mode or FORWARDING mode. In SERVER mode the router will not route IP packets, but will provide Telnet services, respond to SNMP requests, and use TFTP to download software upgrades. In FORWARDING mode the router will route IP

packets, as well as performing all the functions of SERVER mode. The default operational mode is FORWARDING.

See Also DISABLE IP
 DISABLE IP FORWARDING
 DISABLE IP SRCROUTE
 ENABLE IP
 ENABLE IP SRCROUTE
 SHOW IP

ENABLE IP HELPER

Syntax `ENABLE IP HELPER`

Description This command enables the forwarding of broadcast UDP traffic on specified UDP ports to specified destination IP addresses.

Examples To enable broadcast forwarding, use the command:

```
ENABLE IP HELPER
```

See Also ADD IP HELPER
 DELETE IP HELPER
 DISABLE IP HELPER
 SHOW IP HELPER

ENABLE IP INTERFACE

Syntax `ENABLE IP INTERFACE=interface`

where:

- *interface* is an interface name formed by concatenating a layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number in the range 0 to 15 (e.g. eth0, ppp1-1). If a logical interface is not specified, 0 is assumed (i.e. 'eth0' is equivalent to 'eth0-0').

Description This command enables a logical IP interface for use by the IP routing module.

The INTERFACE parameter specifies the name of the logical interface to be enabled. The interface must be assigned to the IP routing module and currently disabled.

Examples To enable the first logical IP interface attached to PPP0, use the command:

```
ENABLE IP INT=PPP0-0
```

See Also ADD IP INTERFACE
 DELETE IP INTERFACE
 DISABLE IP INTERFACE
 RESET IP INTERFACE
 SET IP INTERFACE
 SHOW IP INTERFACE

ENABLE IP REMOTEASSIGN

Syntax ENABLE IP REMOTEASSIGN

Description This command enables the remote assignment of IP addresses for unnumbered PPP interfaces. If a PPP interface is created with an IP address of 0.0.0.0, and remote IP address assignment is enabled, during the IP control protocol (IPCP) negotiation process the router will allow the remote PPP peer to set the IP address of the local PPP interface. If the local PPP interface has an IP number other than 0.0.0.0, or if remote IP address assignment is disabled, the router will not allow the remote PPP peer to set the IP address of the local PPP interface.

Examples To enable remote IP address assignment, use the command:

```
ENABLE IP REMOTEASSIGN
```

See Also DISABLE IP REMOTEASSIGN
 SHOW IP

ENABLE IP ROUTE

Syntax ENABLE IP ROUTE {CACHE|COUNT|MULTIPATH}

Description This command enables route caching, route counters or equal cost multipath routing.

The CACHE parameter enables route caching. The cache is enabled by default.

The COUNT parameter enables counting of octets sent and received to and from a network. It is disabled by default.

The MULTIPATH parameter enables equal cost multipath routing whereby if several routes with the same cost, policy and type exist for a destination, all traffic for that destination will be equally shared among the routes.

Examples To enable byte counting for routes, use the command:

```
ENABLE IP ROUTE COUNT
```

See Also DISABLE IP ROUTE
 SHOW IP ROUTE

ENABLE IP SRCROUTE

Syntax `ENABLE IP SRCROUTE`

Description This command enables the forwarding of source-routed IP packets.

When forwarding is enabled, source-routed IP packets will be forwarded by the router as normal. When forwarding disabled, source-routed packets will be discarded by the router. The default is to disable forwarding, and source-routed IP packets will be discarded.

Source routing is rarely used for legitimate purposes, and is a common method used to circumvent packet-filtering firewalls and to masquerade as a trusted host inside the destination network. This command is therefore an extra security feature, which is why source routed packets are discarded by default.

See Also `DISABLE IP`
`DISABLE IP SRCROUTE`
`ENABLE IP`
`ENABLE IP FORWARDING`
`SHOW IP`

PING

Syntax `PING [[IPADDRESS=] ipadd] [DELAY=seconds] [LENGTH=number]
 [NUMBER={number|CONTINUOUS}] [PATTERN=hexnum]
 [SIPADDRESS=ipadd] [SCREENOUTPUT={YES|NO}]
 [TIMEOUT=number] [TOS=number]`

where:

- *ipadd* is an IP address in dotted decimal notation or a host name from the host name table.
- *seconds* is a decimal number in the range 0 to 4294967295.
- *number* is a decimal number.
- *hexnum* is an eight digit hexadecimal number, optionally preceded by the characters "0x".

Description This command can be used to test that a valid path (route) exists to a destination. Packets are sent to the address specified. If the destination host replies, the time taken for the response to be received is recorded, and optionally displayed. The parameters of this command override the defaults set with the SET PING command on page 6-103.



The PING command does not perform domain name server (DNS) lookups. A valid IP address or a host name defined in the host name table must be specified. Hosts can be added to the host name table with the ADD IP HOST command on page 6-52.

The IPADDRESS parameter specifies the destination address for ping packets.

The DELAY parameter specifies the time interval, in seconds, between ping packets. The default is 1 second.

The LENGTH parameter specifies the number of data bytes of the specified pattern to include in the data portion of the ping packet. If LENGTH is not specified then the current default is used.

The NUMBER parameter specifies the number of ping packets to transmit. If NUMBER is not specified then the current default is used. If CONTINUOUS is specified then the TIMEOUT parameter must be set to a value greater than 0, and packets are sent continuously until the STOP PING command on page 6-145 is issued.

The PATTERN parameter specifies the data to use to fill the data portion of the ping packet. If PATTERN is not specified then the current default is used.

The SIPADDRESS parameter specifies the source address to use in ping packets. If the source address is not specified, and has not been set using the SET PING command on page 6-103, the default is to use the address of the interface from which the ping packets are transmitted. In the special case of IP addresses, the router's local interface IP address, if set, is used. Otherwise the IP address of the interface from which the ping packets are transmitted is used.

The SCREENOUTPUT parameter specifies whether or not the output is sent to the terminal. If YES is specified, the response time for each *Echo Reply* packet is displayed to the terminal as the reply is received from the destination host (Figure 6-13 on page 6-83).

Figure 6-13: Example output from the PING command when SCREENOUTPUT is set to YES.

```
Echo reply 1 from 172.16.8.2 time delay 20 ms
Echo reply 2 from 172.16.8.2 time delay 40 ms
Echo reply 3 from 172.16.8.2 time delay 0 ms
Echo reply 4 from 172.16.8.2 time delay 0 ms
Echo reply 5 from 172.16.8.2 time delay 60 ms
```

If NO is specified, the results are stored and not displayed. To view the results, use the SHOW PING command on page 6-137. If SCREENOUTPUT is not specified then the current default is used.

The TIMEOUT parameter specifies the length of time to wait for a response to a ping packet. If TIMEOUT is not specified then the current default is used. If TIMEOUT is set to zero, then successive ping packets will be transmitted immediately without waiting for a response to previous ping packets.

The TOS parameter is only valid for IP addresses, and specifies the value of the TOS (*Type Of Service*) field in the IP header of the ping packet, as a decimal number in the range 0 to 255. If TOS is not specified then the default is used.

See Also ADD IP HOST
 SET PING
 SHOW PING
 STOP PING

PURGE BOOTP RELAY

Syntax PURGE BOOTP RELAY

Description This command purges the BOOTP configuration. The BOOTP module is disabled, all configuration data (including nonvolatile storage) is purged, and then BOOTP is re-enabled with the default settings.

See Also ADD BOOTP RELAY
DELETE BOOTP RELAY
DISABLE BOOTP RELAY
ENABLE BOOTP RELAY
SET BOOTP MAXHOPS
SHOW BOOTP RELAY

PURGE IP

Syntax PURGE IP

Description This command purges all configuration information relating to the IP routing module, and reinitialises the data structures used by the IP module. It should be used when first setting up the IP module or when a major change is required.



All current configuration information will be lost. Use with extreme caution!

Minor changes, such as changing the IP address of an interface, can be done without using the PURGE IP command. The configuration information is kept in nonvolatile storage so that information will be retained after a power down.

See Also RESET IP

RESET IP

Syntax RESET IP

Description This command reinitialises all dynamic IP data structures. It will not make the router operational if incorrect or incomplete information (e.g. no IP address assigned to an interface) has been entered. It will restart all routing timers and clear the ARP cache and the route table.

This command should not be required during normal operation of the router. There are only two circumstances where a restart of the IP module is required. The first case is where a change is made to one of the interfaces assigned to the IP module, using the ADD IP INTERFACE command on page 6-53, the DELETE IP INTERFACE command on page 6-66 or the SET IP INTERFACE command on page 6-92. In this case the relevant command will automatically restart the IP module, and a manual restart using the RESET IP command is not required. The other situation is where an underlying interface (e.g. a PPP

interface) has changed. In this case the IP module needs to be manually restarted so that it can discover the changes.



This command will cause a message to be sent to the Logging Facility, if one has been defined.

See Also PURGE IP
RESET IP COUNTER
RESET IP INTERFACE

RESET IP COUNTER

Syntax RESET IP COUNTER={ALL|GENERAL|ICMP|INTERFACE|ROUTE|UDP}

Description This command resets the specified group of IP counters to zero (0). The COUNTER parameter specifies the group of counters to be reset. If ALL is specified, all IP counters are reset. If GENERAL, ICMP, INTERFACE, ROUTE or UDP is specified then the general IP, ICMP, IP interface, IP route or UDP counters, respectively, are reset.

Examples To reset the IP route counters to zero, use the command:

```
RESET IP COUNTER=ROUTE
```

See Also RESET IP
RESET IP INTERFACE
SHOW IP COUNTER

RESET IP INTERFACE

Syntax RESET IP INTERFACE=*interface*

where:

- *interface* is an interface name formed by concatenating a layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number in the range 0 to 15 (e.g. eth0, ppp1-1). If a logical interface is not specified, 0 is assumed (i.e. 'eth0' is equivalent to 'eth0-0').

Description This command resets a logical IP interface. All routes associated with the interface, except interface and static routes, are purged from the routing table. All ARPs associated with this interface are purged from the ARP cache, and all counters for this interface are reset to zero (0).

The INTERFACE parameter specifies the name of the logical interface to be reset. The interface must currently be assigned to the IP routing module.

Examples To reset the first logical IP interface attached to PPP0, use the command:

```
RESET IP INT=PPP0
```

See Also ADD IP INTERFACE
 DELETE IP INTERFACE
 DISABLE IP INTERFACE
 ENABLE IP INTERFACE
 RESET IP
 RESET IP COUNTER
 SET IP INTERFACE
 SHOW IP INTERFACE

SET BOOTP MAXHOPS

Syntax SET BOOTP MAXHOPS=1..16

Description This command sets the hop count threshold for discarding BOOTP messages. When the 'hops' field in a BOOTP message exceeds the threshold the BOOTP message is discarded. The hop count in a BOOTP message is incremented each time the message is forwarded by a router. The default value of the threshold is 4. The threshold may be set to any value in the range 1 to 16.

See Also ADD BOOTP RELAY
 DELETE BOOTP RELAY
 DISABLE BOOTP RELAY
 ENABLE BOOTP RELAY
 PURGE BOOTP RELAY
 SHOW BOOTP RELAY

SET IP ARP

Syntax SET IP ARP=*ipadd* INTERFACE=*interface*
 {CIRCUIT=*miox-circuit*|ETHERNET=*macadd*}

where:

- *ipadd* is an IP address in dotted decimal notation.
- *interface* is an interface name formed by concatenating a layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number in the range 0 to 15 (e.g. eth0, ppp1-1). If a logical interface is not specified, 0 is assumed (i.e. 'eth0' is equivalent to 'eth0-0').
- *miox-circuit* is the name of a MIOX circuit defined for an X.25 interface, 1 to 15 characters in length. The name is not case-sensitive.
- *macadd* is the physical Ethernet (MAC) address of a host.

Description This command modifies a static ARP entry in the ARP cache. The ARP entry must already exist.

The ARP parameter specifies the IP address for the ARP entry to modify. The INTERFACE parameter specifies the interface over which the ARP entry can be reached. The specified interface must already exist. The ARP and INTERFACE parameters identify the ARP entry to be modified.

The CIRCUIT and ETHERNET parameters specify the physical address and physical address type for the host. Only one of CIRCUIT or ETHERNET may be specified.

Examples To add an ARP entry for a host with an Ethernet address of 00000008319F and an IP address of 172.16.9.197 on interface eth0, use:

```
ADD IP ARP=172.16.9.197 INT=ETH0 ETHERNET=00000008319F
```

To change the ARP entry for host 172.16.9.197 on interface eth0 (because, for example, the Ethernet interface on the host has been replaced and the host now has an Ethernet address of 00BC00032F9B), use:

```
SET IP ARP=172.16.9.197 INT=ETH0 ETHERNET=00BC00032F9B
```

See Also ADD IP ARP
DELETE IP ARP
SHOW IP ARP

SET IP AUTONOMOUS

Syntax SET IP AUTONOMOUS=1..65535

Description This command sets the router's autonomous system number.



Always use an assigned autonomous system number rather than inventing one.

Autonomous system numbers are assigned by the DDN Network Information Centre (see “Background Reading” on page xxxiii of *Preface* for address details).

See Also SHOW IP

SET IP FILTER

Syntax SET IP FILTER=*filter-number* ENTRY=*entry-number*
 [SOURCE=*ipadd*] [SMASK=*ipadd*] [SPORT={*port-name* |
port-id}] [DESTINATION=*ipadd* [DMASK=*ipadd*]]
 [DPORT={*port-name* | *port-id*}] [ICMPCODE={*icmp-code-name* |
icmp-code-id}] [ICMP TYPE={*icmp-type-name* | *icmp-type-id*}]
 [LOG={4..1600 | DUMP | HEADER | NONE}] [OPTIONS={YES | NO}]
 [PROTOCOL={*protocol* | ANY | EGP | ICMP | OSPF | TCP | UDP}]
 [SESSION={ANY | ESTABLISHED | START}] [SIZE=*size*]
 {ACTION={INCLUDE | EXCLUDE} | POLICY=0..15 | PRIORITY=P0..P7}

where:

- *filter-number* is a number in the range 0 to 299.
- *ipadd* is an IP address in dotted decimal notation.
- *port-name* is the predefined name of an IP port.
- *port-id* is an IP port number, or a range of ports in the form *low:high*.
- *icmp-code-name* is the predefined name for an ICMP reason code.
- *icmp-code-id* is the number of an ICMP reason code.
- *icmp-type-name* is the predefined name of an ICMP message type.
- *icmp-type-id* is the number of an ICMP message type.
- *protocol* is an IP protocol number.
- *size* is a number in the range 64 to 65535.
- *entry-number* is the position of this entry in the filter.

Description This command changes a pattern in an IP traffic filter, policy filter or priority filter.

The FILTER parameter specifies the number of the filter in which the pattern is to be changed. The ENTRY parameter specifies the entry number in the filter to be changed. Filters with numbers in the range 0 to 99 are treated as traffic filters, and use the ACTION parameter to specify the action to take with a packet that matches the pattern. Filters with numbers in the range 100 to 199 are treated as policy filters, and use the POLICY parameter to specify the policy to use when routing a packet that matches the pattern. Filters with numbers in the range 200 to 299 are treated as priority filters, and use the PRIORITY parameter to specify the priority to assign to a packet that matches the pattern.

An interface may have a maximum of one traffic filter, one policy filter and one priority filter, but the same traffic, policy or priority filter can be assigned to more than one interface. Traffic filters are applied to packets received via the interface, whereas policy and priority filters are applied to packets as they are transmitted.

The SOURCE parameter specifies the source IP address, in dotted decimal notation, for the pattern.

The SMASK parameter specifies the mask, in dotted decimal notation, to apply to source addresses for this pattern. The mask is used to determine the portion of the source IP address in the IP packet that is significant for comparison with this pattern. The values of SOURCE and SMASK must be compatible. For each bit in SMASK which is set to zero (0) the equivalent bit in SOURCE must also

be zero (0). If either SOURCE or SMASK is 0.0.0.0, both must be 0.0.0.0. The default is 255.255.255.255.

The SPORT parameter specifies the port to check against the source port for this pattern, as the recognised name of a well-known UDP or TCP port (Table 6-9 on page 6-46, ADD IP FILTER command on page 6-45), a decimal value in the range 0 to 65535, or a range of numbers in the form *low:high*. If *low* is omitted, 0 is assumed. If *high* is omitted, the maximum port number is assumed. If a port other than ANY is specified, the PROTOCOL parameter must also be specified, and must be one of TCP or UDP. The default for SPORT is ANY.

The DESTINATION parameter specifies the destination IP address, in dotted decimal notation, for the pattern. The default is 0.0.0.0.

The DMASK parameter specifies the mask, in dotted decimal notation, to apply to the destination address for this pattern. The mask is used to determine the portion of the destination IP address in the IP packet that is significant for comparison with this pattern. The values of DESTINATION and DMASK must be compatible. For each bit in DMASK which is set to zero (0) the equivalent bit in DESTINATION must also be zero (0). If either DESTINATION or DMASK is 0.0.0.0, both must be 0.0.0.0. If DESTINATION is specified, the default value for DMASK is 255.255.255.255. If DESTINATION is not specified, the default value for DMASK is 0.0.0.0. If DMASK is specified, DESTINATION must also be specified.

The DPORT parameter specifies the port to check against the destination port for this pattern, as the recognised name of a well-known UDP or TCP port (Table 6-9 on page 6-46, ADD IP FILTER command on page 6-45), a decimal value in the range 0 to 65535, or a range of numbers in the form *low:high*. If *low* is omitted, 0 is assumed. If *high* is omitted, the maximum port number is assumed. If a port other than ANY is specified, the PROTOCOL parameter must also be specified, and must be one of TCP or UDP. The default for DPORT is ANY.

The ICMPCODE parameter specifies the ICMP message reason code to match against the ICMP code field in an ICMP packet, as a decimal value in the range 0 to 65535, or the recognised name of an ICMP reason code (Table 6-11 on page 6-48, ADD IP FILTER command on page 6-45). This parameter is only valid when the PROTOCOL parameter is set to ICMP.

The ICMPTYPE parameter specifies the ICMP message type to match against the ICMP type field in an ICMP packet header, as a decimal value in the range 0 to 65535, or the recognised name of an ICMP type (Table 6-10 on page 6-47, ADD IP FILTER command on page 6-45). This parameter is only valid when the PROTOCOL parameter is set to ICMP.

The LOG parameter specifies whether or not any matches to a filter entry result in a log message being sent to the router's logging facility. This parameter enables logging of the IP packet filtering process down to the level of an individual filter entry. If a number in the range 4 to 1600 is specified, the filter number, entry number and IP header information (source and destination IP addresses, protocol, source and destination ports, and size) is logged with a message type/subtype of IPFIL/PASS (for patterns with an INCLUDE action) or IPFIL/FAIL (for patterns with an EXCLUDE action). In addition, the first 4 to 1600 octets of the data portion of TCP, UDP and ICMP packets or the first 4 to 1600 octets after the IP header of other protocol packets are logged with a message type/subtype of IPFIL/DUMP. If DUMP is specified, the filter number, entry number and IP header information (source and destination IP

addresses, protocol, source and destination ports, and size) is logged with a message type/subtype of IPFIL/PASS (for patterns with an INCLUDE action) or IPFIL/FAIL (for patterns with an EXCLUDE action). In addition, the first 32 octets of the data portion of TCP, UDP and ICMP packets or the first 32 octets after the IP header of other protocol packets are logged with a message type/subtype of IPFIL/DUMP. If HEADER is specified, the filter number, entry number and IP header information (source and destination IP addresses, protocol, source and destination ports, and size) is logged with a message type/subtype of IPFIL/PASS (for patterns with an INCLUDE action) or IPFIL/FAIL (for patterns with an EXCLUDE action). If NONE is specified, matches to the filter entry are not logged. The default is NONE.

The OPTIONS parameter specifies the presence or absence of the IP options field to check against for this pattern. If YES is specified, the pattern matches IP packets with options set. If NO is specified, the pattern matches IP packets without options set. The default is to match IP packets with or without IP options set.

The PROTOCOL parameter specifies the protocol to check against the protocol for this pattern, as a decimal value in the range 0 to 65535, or the recognised name of an IP protocol type (Table 6-12 on page 6-49). If either SPORT or DPORT are specified, PROTOCOL must be defined as TCP or UDP. Specifying TCP or UDP will filter out packets from companion protocols, such as ICMP, RIP and OSPF, that do not use TCP or UDP as a transport mechanism. The default is ANY.

The SESSION parameter specifies the type of TCP packet to match, and can only be used when the PROTOCOL parameter specifies TCP. If START is specified, the pattern matches TCP packets with the SYN bit set and the ACK bit clear. If ESTABLISHED is specified, the pattern matches TCP packets with either the SYN bit clear or the ACK bit set. If ANY is specified, the pattern matches any TCP packet. The default is ANY.

The SIZE parameter specifies the maximum reassembled size to match against, for each IP fragment. If the fragment's offset plus size is greater than the value specified, the fragment is discarded.

The ACTION parameter specifies, for traffic filters, the action to take when the pattern is matched. If INCLUSION is specified, the IP packet will be processed and forwarded. If EXCLUSION is specified, the IP packet will be discarded. The ACTION, POLICY and PRIORITY parameters are mutually exclusive—only one may be specified.

The POLICY parameter specifies, for policy filters, the routing policy to use to route packets when the pattern is matched. For policy numbers in the range 0 to 7, only routes with a matching policy will be considered. For policy numbers in the range 8 to 15, only routes with a policy of $n-8$ (where n is the filter policy) will be considered, and the policy value $n-8$ will be written into the TOS field of the packet. The policy number is assigned to incoming packets, but employed during forwarding (transmission). The ACTION, POLICY and PRIORITY parameters are mutually exclusive—only one may be specified.

The PRIORITY parameter specifies, for priority filters, the priority to apply to forwarding packets when the pattern is matched. A low value (P0) assigns a high priority to the packet. A high value (P7) assigns a low priority to the packet. The priority number is assigned to incoming packets, but employed during forwarding (transmission). The ACTION, POLICY and PRIORITY parameters are mutually exclusive—only one may be specified.

Examples To set the session to be matched by entry 3 of filter 2 to ESTABLISHED, use:

```
SET IP FILTER=2 ENTRY=3 SESS=ESTA
```

See Also ADD IP FILTER
ADD IP ROUTE FILTER
DELETE IP FILTER
DELETE IP ROUTE FILTER
SHOW IP FILTER
SHOW IP ROUTE FILTER

SET IP HOST

Syntax SET IP HOST=*name* IPADDRESS=*ipadd*

where:

- *name* is a character string up to 60 characters in length. If the string contains spaces it must be enclosed in double quotes.
- *ipadd* is an IP address in dotted decimal notation.

Description This command modifies the IP address associated with a user-defined name for an IP host in the host name table. The host name table makes it easier to Telnet to commonly accessed hosts by enabling the user to enter a shorter, easier to remember name for the host rather than the host's full IP address or domain name.

The HOST parameter specifies the user-defined name for the IP host. A host with the same name must already exist in the host name table. When a host name is specified in the Telnet command, the entire name will be used to match a name in the host name table. All characters are used in the comparison, including nonalphanumeric characters if they are present.

The IPADDRESS parameter specifies the IP address for the host.

Examples To change the IP address for host name "zaphod" in the host name table from 172.16.1.5 to 172.16.9.8, use:

```
SET IP HOST=Zaphod IP=172.16.9.8
```

To Telnet to the host, use any of the following commands:

```
TELNET zaphod  
TELNET zaphod.company.com  
TELNET 172.16.9.8
```

See Also ADD IP HOST
DELETE IP HOST
SET IP NAMESERVER
SET IP SECONDARYNAMESERVER
SHOW IP HOST

SET IP INTERFACE

Syntax SET IP INTERFACE=*interface* [BROADCAST={0|1}]
 [DIRECTEDBROADCAST={YES|NO|ON|OFF}] [FILTER={0..99|
 NONE}] [FRAGMENT={YES|NO}] [IPADDRESS=*ipadd*|DHCP]
 [MASK=*ipadd*] [METRIC=1..16] [MULTICAST={OFF|SEND|
 RECEIVE|BOTH|ON}] [POLICYFILTER={100..199|NONE}]
 [PRIORITYFILTER={200..299|NONE}] [PROXYARP={ON|OFF}]
 [RIPMETRIC=1..16] [VJC={ON|OFF}]

where:

- *interface* is an interface name formed by concatenating a layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number in the range 0 to 15 (e.g. eth0, ppp1-1). If a logical interface is not specified, 0 is assumed (i.e. 'eth0' is equivalent to 'eth0-0').
- *ipadd* is an IP address in dotted decimal notation.

Description This command modifies the configuration of a logical interface used by the IP module. The INTERFACE parameter specifies the name of the logical interface, and implicitly, the attached layer 2 interface. The interface must currently be assigned to the IP module. At least two interfaces must be defined before the router can route IP packets, but only one interface (usually Ethernet) needs to be defined if the router is acting only as a server. A maximum of 512 interfaces can be added. When an interface is added it is automatically enabled. Only one logical interface may be configured to the same IP network or subnet.

The BROADCAST parameter specifies whether or not an all 1's or all 0's broadcast address will be used. The default is 1.



The BROADCAST parameter should not be set to '0' without careful consideration of the consequences. It is provided to allow compatibility with certain older host implementations which do not meet the current standard.

The DIRECTEDBROADCAST parameter specifies whether or not the router allows network or subnet broadcasts to be forwarded to the network directly attached to the logical interface. The default is NO.

The FILTER parameter specifies the traffic filter to apply to IP packets transmitted or received over the logical interface. The filter must already have been defined with the ADD IP FILTER command on page 6-45. A logical interface may have a maximum of one traffic filter, one policy filter and one priority filter, but the same traffic, policy or priority filter can be assigned to more than one interface. Traffic filters are applied to packets received via the logical interface. The default is not to apply a filter.

The FRAGMENT parameter specifies whether or not the "Do not fragment" bit will be obeyed for outgoing IP packets that are larger than the MTU of the interface. If YES is specified, the "Do not fragment" bit will be ignored and outgoing IP packets that are larger than the MTU of the interface will be fragmented. This is particularly useful for interfaces configured with encapsulation which can potentially increase packet sizes beyond the MTU of the interface. If NO is specified, the "Do not fragment" bit will be obeyed and IP packets that are larger than the MTU of the interface will be discarded. This is the normal behaviour for IP. The FRAGMENT parameter has no effect on the processing of packets smaller than the interface MTU. The default is NO.

The IPADDRESS parameter specifies the IP address of the logical interface. If DHCP is specified, the router will act as a DHCP client and obtain the configuration of the IP interface via DHCP. Table 6-13 on page 6-54 lists the parameters from the DHCP reply used by the router. If an IP interface is configured to use DHCP to obtain its IP address and subnet mask, the interface will not take part in IP routing until the IP address and subnet mask have been set by DHCP.



Remote address assignment must be enabled using the `ENABLE IP REMOTEASSIGN` command before IP interfaces will accept addresses dynamically assigned by DHCP.

The MASK parameter specifies the subnet mask for the logical interface. The value must be consistent with the value specified for the IPADDRESS parameter. The default is the network mask for the address class of the IP address (e.g. 255.255.0.0 for a Class B address, 255.255.255.0 for a Class C address). If IPADDRESS is set to DHCP, the MASK parameter should not be set as the subnet mask received from the DHCP server will be used.

The MULTICAST parameter specifies how the router will handle multicast packets. If OFF is specified, the router will neither send nor receive multicast packets. If SEND is specified, the router will send but not receive multicast packets. If RECEIVE is specified, the router will receive but not send multicast packets. If BOTH is specified the router will both send and receive multicast packets. The value ON is a synonym for BOTH. Note that this parameter applies to the entire IP interface, not an individual logical interface. Setting the MULTICAST parameter on one logical interface will set the MULTICAST parameter on all other logical interfaces associated with the same IP interface. The default is RECEIVE.

The POLICYFILTER parameter specifies the policy filter to apply to IP packets received over the logical interface. The filter must already have been defined with the ADD IP FILTER command on page 6-45. A logical interface may have a maximum of one traffic filter, one policy filter and one priority filter, but the same traffic, policy or priority filter can be assigned to more than one interface. Policy filters are applied to packets as they are transmitted. The default is not to apply a filter.

The PRIORITYFILTER parameter specifies the priority filter to apply to IP packets received over the logical interface. The filter must already have been defined with the ADD IP FILTER command on page 6-45. A logical interface may have a maximum of one traffic filter, one policy filter and one priority filter, but the same traffic, policy or priority filter can be assigned to more than one interface. Priority filters are applied to packets as they are transmitted. The default is not to apply a filter.

The PROXYARP parameter enables or disables proxy ARP responses to ARP requests. This parameter is only valid for Ethernet interfaces. The default is ON.

The RIPMETRIC parameter specifies the cost of crossing the logical interface, for RIP. The default is 1. The METRIC parameter is also accepted, for backward compatibility.

The VJC parameter is only valid for Point-to-Point Protocol (PPP) and X25T interfaces, and specifies whether or not Van Jacobson header compression is to be used on the layer 2 interface. The VJC parameter applies to all logical interfaces attached to the same layer 2 interface. Changing the setting on one

logical interface will alter the setting on all other logical interfaces attached to the layer 2 interface. Compression provides the most advantage on slower link speeds (up to 48 kbps). At speeds of 64 kbps and higher, compression will actually reduce efficiency and so should be disabled. The default is OFF.



Van Jacobson's TCP/IP header compression should not be enabled on a multilink PPP interface.



For security reasons this command will only be accepted if the user has SECURITY OFFICER privilege.

Examples To set the first IP interface attached to PPP2 with an IP address of 172.16.248.33, a subnet mask of 255.255.255.0, and a metric of 5, use the command:

```
SET IP INT=PPP2-0 IP=172.16.248.33 MASK=255.255.255.0
RIPMET=5
```

See Also ADD IP INTERFACE
DELETE IP INTERFACE
DISABLE IP INTERFACE
ENABLE IP INTERFACE
RESET IP INTERFACE
SHOW IP INTERFACE

SET IP LOCAL

Syntax SET IP LOCAL [FILTER={*filter-number*|NONE}]
[IPADDRESS=*ipadd*] [POLICYFILTER={*filter-number*|NONE}]
[PRIORITYFILTER={*filter-number*|NONE}]

where:

- *filter-number* is a number in the range 0 to 299.
- *ipadd* is an IP address in dotted decimal notation.

Description This command modifies the configuration of the router's local IP interface. The local IP interface is a virtual interface which represents the IP routing module itself. The interface can be assigned an IP address, which can then be used as the source address of IP packets generated internally by IP protocols such as PING. Higher layer protocols such as PING must assign a source IP address to packets passed to IP for forwarding. The following rules are used to determine which IP address to use as the source address:

1. If the higher layer protocol's configuration specifies a source IP address to use, then the configured address is used as the packet's source IP address.
For example, the SIPADDRESS parameter of the PING command on page 6-82 specifies the source IP address to use in ping packets.
2. If the local IP interface has been assigned an IP address, then the IP address of the local interface is used as the packet's source IP address.

3. Otherwise, the IP routing module determines the interface over which the packet is to be transmitted, and assigns the IP address of the interface as the packet's source IP address.

The FILTER parameter specifies the filter to apply to IP packets transmitted or received over the interface. The filter must already have been defined with the ADD IP FILTER command on page 6-45. An interface may have a maximum of one traffic filter, one policy filter and one priority filter, but the same traffic, policy or priority filter can be assigned to more than one interface. Traffic filters are applied to packets received via the interface. The default is not to apply a filter.

The IPADDRESS parameter specifies the IP address of the interface. The IP address must be the IP address of one of the router's active IP interfaces. Specifying an IP address of 0.0.0.0 effectively 'unsets' the IP address of the local interface.

The POLICYFILTER parameter specifies the policy filter to apply to IP packets received over the interface. The filter must already have been defined with the ADD IP FILTER command on page 6-45. An interface may have a maximum of one traffic filter, one policy filter and one priority filter, but the same traffic, policy or priority filter can be assigned to more than one interface. Policy filters are applied to packets as they are transmitted. The default is not to apply a filter.

The PRIORITYFILTER parameter specifies the priority filter to apply to IP packets received over the interface. The filter must already have been defined with the ADD IP FILTER command on page 6-45. An interface may have a maximum of one traffic filter, one policy filter and one priority filter, but the same traffic, policy or priority filter can be assigned to more than one interface. Priority filters are applied to packets as they are transmitted. The default is not to apply a filter.

Examples To set the IP address of the local IP interface to 192.168.33.11, use:

```
SET IP LOCAL IP=192.168.33.11
```

To remove the IP address of the local IP interface, use:

```
SET IP LOCAL IP=0.0.0.0
```

See Also ADD IP INTERFACE
DELETE IP INTERFACE
SET IP INTERFACE
SHOW IP INTERFACE

SET IP NAMESERVER

Syntax SET IP NAMESERVER=*ipadd*

where:

- *ipadd* is an IP address in dotted decimal notation.

Description This command specifies the IP address of a host able to act as the primary name server for the router. Name servers are used to resolve Telnet requests to

host names which are not in the host name table. If the host is entered into the host table, then no access to a name server will be required. This may suit installations which have no name server.

A secondary name server can also be specified with the SET IP SECONDARYNAMESEVER command on page 6-102. When the router performs a DNS lookup, it firsts sends the request to the primary name server. If a response is not received within 20 seconds the request is sent to the secondary name server.

Examples To specify the host with IP address 172.16.1.5 as a name server, use:

```
SET IP NAMESERVER=172.16.1.5
```

See Also ADD IP HOST
DELETE IP HOST
SET IP HOST
SET IP SECONDARYNAMESEVER
SHOW IP
SHOW IP HOST

SET IP RIP

Syntax SET IP RIP INTERFACE=*interface* [CIRCUIT=*miox-circuit*]
[IP=*ipadd*] [SEND={NONE|RIP1|RIP2|COMPATIBLE}]
[RECEIVE={NONE|RIP1|RIP2|BOTH}] [DEMAND={NO|YES}]
[AUTH={NONE|PASSWORD|MD5}] [PASSWORD=*password*]

where:

- *interface* is an interface name formed by concatenating a layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number in the range 0 to 15 (e.g. eth0, ppp1-1). If a logical interface is not specified, 0 is assumed (i.e. 'eth0' is equivalent to 'eth0-0').
- *miox-circuit* is the name of a MIOX circuit defined for an X.25 interface, 1 to 15 characters in length. The name is not case-sensitive.
- *ipadd* is an IP address in dotted decimal notation.
- *password* is a character string, 1 to 15 characters in length. It may contain letters (a-z), digits (0-9) and the underscore character ("_").

Description This command sets attributes of the RIP neighbour. The IP address and the interface identify which RIP neighbour to change.

The INTERFACE parameter specifies the interface the RIP neighbour is on.

The CIRCUIT parameter specifies the X.25 circuit on which to send or receive RIP packets. It is a required parameter for X25T interfaces and is valid only when the interface is an X25T interface.

The IP parameter specifies the IP address of the RIP neighbour.

The SEND parameter specifies the version of RIP packet to send. If NONE is specified then no RIP packets will be sent. If RIP1 is specified then RIP version 1 packets are sent. If RIP2 is specified then RIP version 2 packets are sent. If

COMPATIBLE is specified RIP version 2 packets are sent without routes that a router receiving only RIP version 1 will treat as host routes. The default is RIP1.

The RECEIVE parameter specifies the version of RIP packets to receive. If NONE is specified then no RIP packets are accepted from the IP address on the interface. If RIP1 is specified then only RIP version 1 packets are accepted. If RIP2 is specified then only RIP version 2 packets are accepted. If BOTH is specified then either RIP version 1 or RIP version 2 packets are accepted. The default is BOTH.

The DEMAND parameter specifies whether to use the RIP demand procedures when send and receiving RIP and for routes received from this neighbour. If NO is specified the demand procedures are not used. If YES is specified the demand procedures are used. The default is NO.

The AUTHENTICATION parameter specifies the method used to authenticate RIP packets. This must be NONE unless using RIP version 2. If NONE is specified no authentication is used. If PASSWORD is specified then a clear text password is used to authenticate RIP packets. If MD5 is specified then an encrypted password is used to authenticate a RIP packet. The default is NONE.

The PASSWORD parameter specifies the password to use if the AUTHENTICATION parameter is set to PASSWORD or MD5. This parameter is required if authentication is used.

Examples To change the password for a RIP neighbour using authentication, use the command:

```
SET IP RIP INTERFACE=ppp0 IP=172.16.248.33
PASSWORD=bobsyouruncle
```

To change a RIP neighbour from on-demand using RIP version 2 to not on-demand sending RIP version 1 compatible packets, and receiving RIP version 1 and 2, use the command:

```
SET IP RIP INTERFACE=ppp0 IP=172.16.248.33 DEMAND=NO
SEND=COMPATIBLE RECEIVE=BOTH
```

See Also ADD IP RIP
DELETE IP RIP
SET IP RIPTIMER
SHOW IP RIP

SET IP RIPTIMER

Syntax SET IP RIPTIMER [FLUSH=*seconds*] [HOLDDOWN=*seconds*]
[INVALID=*seconds*] [UPDATE=*seconds*]

where:

■ *seconds* is a decimal number in the range 1 to 4294967295.

Description This command sets the values of the global RIP timers, in seconds.

The UPDATE parameter sets the time interval (in seconds) between RIP updates for all interfaces not using RIP on demand. The default is 30.

The **INVALID** parameter sets the time interval (in seconds) after which the router will deem a route to be invalid if no update has been received for the route. The default is 180.

The **HOLDDOWN** parameter sets the time interval (in seconds), after a route has become invalid, during which the router will ignore updates for the route which would normally make the route valid again. The default is 120.

The **FLUSH** parameter sets the time interval (in seconds), from the last update of a route, until the route is flushed from the route table. This time should be set equal to or higher than the sum of the **INVALID** and **HOLDDOWN** times. The default is 300.

Examples To force RIP routes to be invalidated and flushed as soon as a single update is missed, use the command:

```
SET IP RIPTIMER INVALID=35 HOLDDOWN=0 FLUSH=35
```

See Also SET IP RIP
SHOW IP RIP
SHOW IP RIPTIMER

SET IP ROUTE

Syntax SET IP ROUTE=*ipadd* INTERFACE=*interface* MASK=*ipadd*
NEXTHOP=*ipadd* [CIRCUIT=*miox-circuit*] [METRIC=1..16]
[METRIC1=1..16] [POLICY=0..7] [PREFERENCE=0..65535]

where:

- *ipadd* is an IP address in dotted decimal notation.
- *interface* is an interface name formed by concatenating a layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number in the range 0 to 15 (e.g. eth0, ppp1-1). If a logical interface is not specified, 0 is assumed (i.e. 'eth0' is equivalent to 'eth0-0').
- *miox-circuit* is the name of a MIOX circuit defined for an X.25 interface, 1 to 15 characters in length. The name is not case-sensitive.

Description This command modifies a static route in the IP route table. Static routes can be used to define default routes to external routers or networks. A default route is one with a network address of 0.0.0.0. If the router receives data and can not find a route for it, the data will be sent to the default route. To define a default route IPADDRESS is set to 0.0.0.0 and NEXTHOP points to the network (router) to which default packets are to be directed. The static route must not already exist. However, if the route exists as a dynamic (e.g. RIP-derived) route, the static route may still be added. A maximum of 300 static routes can be defined.

This command is also used to define subnets. Multiple routes can be defined for a single interface (usually a LAN). This is useful if it is desired to have more than one network or subnet present on a particular interface. A common reason is growth in the number of hosts exceeding the capacity of a single subnet. Additional subnets can be assigned by adding static routes. In this case IPADDRESS is set to the new subnet, NEXTHOP is set to 0.0.0.0, and METRIC should be set to 1.

The ROUTE parameter specifies the IP address of the static route.

The INTERFACE parameter specifies the IP interface with which the route is associated. The interface must already exist and be assigned to the IP module. If the interface is an X.25 DTE interface, the CIRCUIT parameter is required and specifies the name of a MIOX circuit already defined for the X.25 DTE interface.

The NEXTHOP parameter specifies the IP address of the next hop (router) for the route. The default is the IP address of the interface specified by the INTERFACE parameter. For a PPP link, NEXTHOP should be the IP address of the remote end of the PPP link.

The MASK parameter specifies the subnet mask for the route. A check is performed on the route and mask to verify that the route is the same before and after masking. This ensures that a static route is not specified to more than its subnet mask.



In most cases the subnet mask will not need to be specified because the method outlined above will work for most common cases.

The METRIC1 parameter specifies the cost of traversing the route for RIP. The default is 1. The normal range is 2 to 16. A metric of 1 should only be used when adding a subnet to an interface. The METRIC parameter is also accepted, for backward compatibility.

The POLICY parameter specifies the type of service for the route. The default is 0.

The PREFERENCE parameter specifies the preference for the route. When more than one route in the route table matches the destination address in an IP packet, the route with the lowest preference value will be used to route the packet. If two or more candidate routes have the same preference, the route with the longest subnet mask will be used. Interface routes have a preference of 0 and RIP routes have a preference of 100. The default preference for static routes other than 0.0.0.0 is 60. The default for the default static route 0.0.0.0 is 360.

Examples To set the subnet 172.16.9.0 on interface ETH0 to have a RIP metric of 2, use the command:

```
SET IP ROUTE=172.16.9.0 MASK=255.255.255.0 INTERFACE=ETH0
NEXTHOP=0.0.0.0 METRIC=2
```

See Also ADD IP ROUTE
DELETE IP ROUTE
SHOW IP ROUTE

SET IP ROUTE FILTER

Syntax SET IP ROUTE FILTER=*filter-id* IP=*ipadd* MASK=*ipadd*
ACTION={ INCLUDE | EXCLUDE } [DIRECTION={ RECEIVE | SEND |
BOTH }] [INTERFACE=*interface*] [NEXTHOP=*ipadd*]
[POLICY=0..7] [PROTOCOL={ ANY | RIP | STATIC | INTERFACE }]

where:

- *filter-id* is a number in the range 1 to 100.
- *ipadd* is an IP address in dotted decimal notation.
- *interface* is an interface name formed by concatenating a layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number in the range 0 to 15 (e.g. eth0, ppp1-1). If a logical interface is not specified, 0 is assumed (i.e. 'eth0' is equivalent to 'eth0-0').

Description This command modifies a route filter. A route filter controls which routes are sent and received by the routing protocols.

The FILTER parameter specifies the index in the filter list of the filter to modify. The specified entry must exist.

The IP parameter specifies the network address to match. The wildcard character ("*") can be used to match a network range. For example, 192.168.*.* will match all destination networks which start with 192.168. The wildcard character can only be used to replace a complete number—192.168.*.* is valid but 192.16.*.* is not valid.

The MASK parameter specifies the network mask of the network to match. The wildcard character ("*") can be used to match a network mask range. For example, 255.255.*.* will match all destination network masks which start with 255.255. The wildcard character can only be used to replace a complete number—255.255.*.* is valid but 255.25.*.* is not valid.

The ACTION parameter specifies what to do with routes that match the filter. If INCLUDE is specified then the route is included. If EXCLUDE is specified then the route is excluded.

The DIRECTION parameter specifies whether to filter the route when receiving it or when sending it.

The INTERFACE parameter specifies the interface to which the filter applies. If specified the route will only be filtered if the route is sent or received on the interface.

The NEXTHOP parameter specifies the IP address of the next hop router to match. If specified the route will only be filtered if the route is sent or received to or from the next hop.

The POLICY parameter specifies the type of service to filter. If not specified all types of service are filtered.

The PROTOCOL parameter specifies the routing protocol to which the filter applies. If specified the route will only be filtered if the route is sent or received by the specified protocol. The default is ANY.

Examples To limit IP route filter 1 to route information received by the router, use the command:

```
ADD IP ROUTE FILT=1 PROT=RIP ACT=INCL DIR=RECEIVE IP=0.0.0.0  
MASK=0.0.0.0
```

See Also ADD IP ROUTE FILTER
DELETE IP ROUTE FILTER
SHOW IP ROUTE FILTER

SET IP ROUTE TEMPLATE

Syntax SET IP ROUTE TEMPLATE=*name* [NEXTHOP=*ipadd*]
[CIRCUIT=*miox-circuit*] [METRIC=1..16] [METRIC1=1..16]
[POLICY=0..7] [PREFERENCE=0..65535]

where:

- *name* is a character string, 1 to 31 characters in length. Valid characters are any printable character. If *name* contains spaces it must be enclosed in double quotes. *name* is not case-sensitive.
- *interface* is an interface name formed by concatenating a layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number in the range 0 to 15 (e.g. eth0, ppp1-1). If a logical interface is not specified, 0 is assumed (i.e. 'eth0' is equivalent to 'eth0-0').
- *ipadd* is an IP address in dotted decimal notation.
- *miox-circuit* is the name of a MIOX circuit defined for an X.25 interface, 1 to 15 characters in length. The name is not case-sensitive.

Description This command modifies an existing IP route template. IP route templates are used by the router software to add IP routes to IP subnetworks discovered during normal operation by other protocols. This is only required if IP traffic to the discovered IP subnetwork needs to be routed via a route other than the default route.

The CIRCUIT parameter specifies the name of a MIOX circuit to use on the X.25 DTE interface. This parameter is only valid if the template was added with a X.25 DTE interface.

The NEXTHOP parameter specifies the IP address of the next hop (router) for any route added using this template. The default is the IP address of the interface specified by the INTERFACE parameter. For a PPP link, NEXTHOP should be the IP address of the remote end of the PPP link.

The METRIC1 parameter specifies the cost of traversing any route added using this template for RIP. The default is 1. The normal range is 2 to 16. A metric of 1 should only be used when adding a subnet to an interface. The METRIC parameter is also accepted, for backward compatibility.

The POLICY parameter specifies the type of service for any route added using this template. The default is 0.

The PREFERENCE parameter specifies the preference for any route added using this template. When more than one route in the route table matches the

destination address in an IP packet, the route with the lowest preference value will be used to route the packet. If two or more candidate routes have the same preference, the route with the longest subnet mask will be used. Interface routes have a preference of 0 and RIP routes have a preference of 100. The default preference for static routes other than 0.0.0.0 is 60. The default for the default static route 0.0.0.0 is 360.

Examples To set the preference of routes created with the IP route template named "branch_office" to 90, use the command:

```
SET IP ROUTE TEMPLATE=branch_office preference=90
```

See Also ADD IP ROUTE TEMPLATE
DELETE IP ROUTE TEMPLATE
SHOW IP ROUTE TEMPLATE

SET IP SECONDARYNAMESESERVER

Syntax SET IP SECONDARYNAMESESERVER=*ipadd*

Where:

■ *ipadd* is an IP address in dotted decimal notation.

Description This command sets the IP address of the secondary name server. Name servers are used to resolve Telnet requests to host names which are not in the host name table. If the host is entered into the host table, then no access to a name server will be required. This may suit installations which have no name server. When the router performs a DNS lookup, it firsts sends the request to the primary name server. If a response is not received within 20 seconds the request is sent to the secondary name server.

Example To set the router's secondary name server address to 192.168.2.1, use the command:

```
SET IP SECONDARYNAMESESERVER=192.168.2.1
```

See Also ADD IP HOST
DELETE IP HOST
SET IP HOST
SET IP NAMESERVER
SHOW IP
SHOW IP HOST

SET PING

Syntax SET PING [[IPADDRESS=] *ipadd*] [DELAY=*seconds*]
 [LENGTH=*number*] [NUMBER={*number*|CONTINUOUS}]
 [PATTERN=*hexnum*] [SIPADDRESS=*ipadd*] [SCREENOUTPUT={YES|
 NO}] [TIMEOUT=*number*] [TOS=*number*]

where:

- *ipadd* is an IP address in dotted decimal notation or a host name from the host name table.
- *seconds* is a decimal number in the range 0 to 4294967295.
- *number* is a decimal number.
- *hexnum* is an eight digit hexadecimal number, optionally preceded by the characters "0x".

Description This command sets the defaults for the PING command on page 6-82.

If there is no default destination and a destination is not specified on the PING command on page 6-82 then a ping is not generated and an error message is displayed.

The IPADDRESS parameter specifies the destination address for ping packets. If the IPADDRESS parameter has previously been set, it can be restored to the default "not set" state by specifying the value 0.0.0.0.

The DELAY parameter specifies the time interval, in seconds, between ping packets. The default is 1 second.

The LENGTH parameter specifies the number of data bytes of the specified pattern to include in the data portion of the ping packet. If LENGTH is not specified then the current default is used.

The NUMBER parameter specifies the number of ping packets to transmit. If NUMBER is not specified then the current default is used. If CONTINUOUS is specified then the TIMEOUT parameter must be set to a value greater than 0, and packets are sent continuously until the STOP PING command on page 6-145 is issued.

The PATTERN parameter specifies the data to use to fill the data portion of the ping packet. If PATTERN is not specified then the current default is used.

The SIPADDRESS parameter specifies the source address to use in ping packets. If the source address is not specified, and has not been set using the SET PING command on page 6-103, the default is to use the address of the interface from which the ping packets are transmitted. In the special case of IP addresses, the router's local interface IP address, if set, is used. Otherwise the IP address of the interface from which the ping packets are transmitted is used. If the SIPADDRESS parameter has previously been set, it can be restored to the default "not set" state by specifying the value 0.0.0.0.

The SCREENOUTPUT parameter specifies whether or not the output is sent to the terminal. If YES is specified, the response time for each *Echo Reply* packet is displayed to the terminal as the reply is received from the destination host (Figure 6-13 on page 6-83). If NO is specified, the results are stored and not displayed. To view the results, use the SHOW PING command on page 6-137. If SCREENOUTPUT is not specified then the current default is used.

The TIMEOUT parameter specifies the length of time to wait for a response to a ping packet. If TIMEOUT is not specified then the current default is used. If TIMEOUT is set to zero, then successive ping packets will be transmitted immediately without waiting for a response to previous ping packets.

The TOS parameter is only valid for IP addresses, and specifies the value of the TOS (*Type Of Service*) field in the IP header of the ping packet, as a decimal number in the range 0 to 255. If TOS is not specified then the current default is used.

See Also ADD IP HOST
 PING
 SHOW PING
 STOP PING

SET TRACE

Syntax SET TRACE [[IPADDRESS=] *ipadd*] [MAXTTL=*number*]
 [MINTTL=*number*] [NUMBER=*number*] [PORT=*port-number*]
 [SCREENOUTPUT={YES|NO}] [SOURCE=*ipadd*] [TIMEOUT=*number*]
 [TOS=*number*]

where:

- *ipadd* is an IP address in dotted decimal notation or a host name from the host name table.
- *number* is a decimal number.
- *port-number* is an IP port number.

Description This command sets default options for the trace route command.

If there is no default destination and a destination is not specified with the TRACE command on page 6-146 then a trace is not performed and an error message is displayed.

The IPADDRESS parameter specifies the destination IP address. The command will trace the route to this IP address.

The MAXTTL parameter specifies the maximum value for the TTL (*Time To Live*) field in the IP packet, and is used to limit the trace route to a maximum number of hops. If MAXTTL is not specified then the current default is used.

The MINTTL parameter specifies the initial value of the TTL (*Time To Live*) field in the IP packet, and can be used to skip some hops at the start of the route. If MINTTL is not specified the current default is used.

The NUMBER parameter specifies the number of packets to send to each hop. If NUMBER is not specified the current default is used. A maximum of 100 packets may be transmitted.

The PORT parameter specifies the UDP destination port number for the packets being transmitted, and can be used to detect whether or not there is an IP device listening on the specified port. If an IP device is listening on the port,

the ICMP “unreachable” message which trace route depends on will not be returned.

The SCREENOUTPUT parameter specifies whether or not the output is sent to the terminal. If SCREENOUTPUT is not specified then the current default is used.

The SOURCE parameter specifies the IP address to use as a source address in the packets. If SOURCE is not specified then the IP address of the interface from which the packets are transmitted is used.

The TIMEOUT parameter specifies the length of time to wait for a response before sending packets to the next hop. If TIMEOUT is not specified then the current default is used. If ICMP “unreachable” messages are received within the timeout period, then packets are transmitted to the next hop immediately.

The TOS parameter specifies the value of the TOS (*Type Of Service*) field in the IP header of the packets being transmitted, as a decimal number in the range 0 to 255. If TOS is not specified then the current default is used.

See Also ADD IP HOST
SHOW TRACE
STOP TRACE
TRACE

SHOW BOOTP RELAY

Syntax SHOW BOOTP RELAY

Description This command displays the current configuration of the BOOTP Relay Agent (Figure 6-14 on page 6-105, Table 6-14 on page 6-106).

Figure 6-14: Example output from the SHOW BOOTP RELAY command.

```

BOOTP Relaying Agent Configuration.

Status      : ENABLED
Maximum Hops : 4

BOOTP Relay Destinations
-----
192.231.35.29
-----

BOOTP Counters
-----
InPackets   OutPackets   InRejects   InRequests   InReplies
0000000000  0000000000  0000000000  0000000000  0000000000

```

Table 6-14: Parameters displayed in the output of the SHOW BOOTP RELAY command.

Parameter	Meaning
Status	The status of the BOOTP Relay Agent; one of "ENABLED" or "DISABLED".
Maximum Hops	The maximum value allowed for the 'hops' field in a BOOTP message, before the message is discarded.
BOOTP Relay Destinations	The list of IP addresses to which BOOTREQUEST messages will be forwarded.
InPackets	The total number of BOOTP packets received.
OutPackets	The total number of BOOTP packets transmitted.
InRejects	The number of incoming BOOTP packets that were rejected because of an error in the packet.
InRequests	The number of BOOTP requests received.
InReplies	The number of BOOTP replies received.

See Also ADD BOOTP RELAY
DELETE BOOTP RELAY
DISABLE BOOTP RELAY
ENABLE BOOTP RELAY
PURGE BOOTP RELAY
SET BOOTP MAXHOPS

SHOW IP

Syntax SHOW IP

Description This command displays general configuration information regarding the router (Figure 6-15 on page 6-107, Table 6-15 on page 6-107).

Figure 6-15: Example output from the SHOW IP command.

```

IP Module Configuration
-----

Module Status ..... Enabled
IP Packet Forwarding ..... Enabled
IP Echo Reply ..... Enabled
Debugging ..... Disabled
IP Fragment Offset Filtering ... Enabled
Name Server ..... Not Set
Secondary Name Server ..... Not Set
Source-Routed Packets ..... Discarded
Remote IP address assignment ... Disabled
DNS Relay ..... Disabled

Routing Protocols

RIP Neighbours ..... 1

Active Routes

Static ..... 0
Interface ..... 1
RIP ..... 4
Other ..... 0

IP Filter Configuration

Total filters ..... 0

```

Table 6-15: Parameters displayed in the output of the SHOW IP command.

Parameter	Meaning
Module Status	The current state of the IP module; one of "Enabled" or "Disabled".
IP Packet Forwarding	The current state of the IP forwarding function; one of "Enabled" (IP module is in FORWARDING mode) or "Disabled" (IP module is in SERVER mode).
IP Echo Reply	Whether or not replies to echo request messages are enabled; one of "Enabled" or "Disabled".
Debugging	The current state of the IP debugging facility; one of "Enabled" or "Disabled".
IP Fragment Offset Filter	The current state of the IP fragment offset filtering; one of "Enabled" or "Disabled".
Name Server	The IP address of the primary name server, or "Not Set" if a name server is not assigned.
Secondary Name Server	The IP address of the secondary name server, or "Not Set" if a secondary name server is not assigned.
Source-Routed Packets	Whether source-routed packets are forwarded or discarded; one of "Forwarded" or "Discarded".
Remote IP address assignment	Whether or not remote IP address assignment is enabled; one of "Enabled" or "Disabled".
DNS Relay	Whether or not the DNS relay agent is enabled; one of "Enabled" or "Disabled".

Table 6-15: Parameters displayed in the output of the SHOW IP command.

Parameter	Meaning
RIP Neighbours	The number of RIP neighbours defined.
Static	The number of static routes in use.
Interface	The number of interface-related routes in use.
RIP	The number of RIP-derived routes in use.
Other	The number of other routes in use.
Filter <i>n</i>	The number of patterns in filter <i>n</i> .
Total Filters	The total number of IP filters defined.

See Also DISABLE IP
 DISABLE IP DEBUG
 DISABLE IP DNSRELAY
 DISABLE IP FORWARDING
 DISABLE IP SRCROUTE
 DISABLE SNMP in *Chapter 16, Simple Network Management Protocol (SNMP)*
 ENABLE IP
 ENABLE IP DEBUG
 ENABLE IP DNSRELAY
 ENABLE IP FORWARDING
 ENABLE IP SRCROUTE
 ENABLE SNMP in *Chapter 16, Simple Network Management Protocol (SNMP)*
 SET IP NAMESERVER
 SET IP SECONDARYNAMESERVER

SHOW IP ARP

Syntax SHOW IP ARP

Description This command displays the contents of the ARP cache. The ARP cache contains mappings of IP addresses to physical addresses for hosts to which the router has recently routed packets. To have an entry in the ARP cache, a host must have attempted to access another host, and it must have found the physical address by using the ARP protocol (Figure 6-16 on page 6-108, Table 6-16 on page 6-109).

Figure 6-16: Example output from the SHOW IP ARP command.

Interface	IP Address	Physical Address	ARP Type	Status
eth0	172.16.8.1	AA-00-04-00-2D-08	Dynamic	Active
eth0	172.16.8.2	AA-00-04-00-28-08	Dynamic	Active
eth0	172.16.8.34	00-00-0C-02-5A-0A	Dynamic	Active
eth0	172.16.9.185	08-03-50-37-00-00	Dynamic	Active
x25t0	172.16.198.1	Remotel	Static	Active
eth0	192.168.163.47	FF-FF-FF-FF-FF-FF	Other	Active
eth0	255.255.255.255	FF-FF-FF-FF-FF-FF	Other	Active

Table 6-16: Parameters displayed in the output of the SHOW IP ARP command.

Field	Meaning
Interface	The interface over which the network device is accessed. If multihoming is enabled (two or more logical interfaces have been assigned to a single layer 2 interface), all interface names will include a hyphen ("-") and the logical interface number.
IP Address	The IP address of the network device.
Physical Address	The physical address of the network device. For an Ethernet, this is the Ethernet address; for a MIOX (X.25) circuit it is the circuit name.
ARP Type	The type of ARP entry; one of "Static" (added using the ADD IP ARP command on page 6-44), "Dynamic" (learned from ARP request/reply message exchanges), "Invalid" (invalid, e.g. the interface currently does not exist), or "Other" (automatically generated by the system; for example, general IP broadcast and IP subnet/network broadcast addresses are added when the IP module is configured).
Status	The current status of the ARP entry; one of "Active" or "Inactive".

See Also ADD IP ARP
DELETE IP ARP
SET IP ARP

SHOW IP COUNTER

Syntax SHOW IP COUNTER [= {ALL | ICMP | INTERFACE | IP | MULTICAST | RIP | ROUTES | UDP }]

Description This command displays all or selected parts of the IP MIB. If an option is not specified, or ALL is specified, all the IP counters are displayed. The MIB can be selectively displayed by specifying one of the options ICMP (Figure 6-17 on page 6-110, Table 6-17 on page 6-110), INTERFACE (Figure 6-18 on page 6-111, Table 6-18 on page 6-111), IP (Figure 6-19 on page 6-112, Table 6-19 on page 6-112), MULTICAST (Figure 6-20 on page 6-113, Table 6-20 on page 6-114), RIP (Figure 6-21 on page 6-114, Table 6-21 on page 6-114), ROUTES (Figure 6-22 on page 6-115, Table 6-22 on page 6-115) or UDP (Figure 6-23 on page 6-115, Table 6-23 on page 6-116).

Figure 6-17: Example output from the SHOW IP COUNTER=ICMP command.

ICMP counters

```

inMsgs ..... 0      outMsgs ..... 0
inErrors ..... 0     outErrors ..... 0
inDestUnreachs ..... 0 outDestUnreachs ..... 0
inTimeExcds ..... 0  outTimeExcds ..... 0
inParamProbs ..... 0  outParamProbs ..... 0
inSrcQuenchs ..... 0  outSrcQuenchs ..... 0
inRedirects ..... 0    outRedirects ..... 0
inEchos ..... 0        outEchos ..... 0
inEchoReps ..... 0     outEchoReps ..... 0
inTimestamps ..... 0   outTimestamps ..... 0
inTimestampReps ..... 0 outTimestampReps ..... 0
inAddrMasks ..... 0    outAddrMasks ..... 0
inAddrMaskReps ..... 0  outAddrMaskReps ..... 0

```

Table 6-17: Parameters displayed in the output of the SHOW IP COUNTER=ICMP command.

Parameter	Meaning
inMsgs	The number of ICMP packets received.
inErrors	The number of ICMP packets received which had ICMP-specific errors (bad ICMP checksums, bad length, etc).
inDestUnreachs	The number of ICMP <i>Destination Unreachable</i> packets received.
inTimeExcds	The number of ICMP <i>Time Exceeded</i> packets received.
inParamProbs	The number of ICMP <i>Parameter Problem</i> packets received.
inSrcQuenchs	The number of ICMP <i>Source Quench Request</i> packets received.
inRedirects	The number of ICMP <i>Redirect Request</i> packets received.
inEchos	The number of ICMP <i>Echo Request</i> (ping) packets received.
inEchoReps	The number of ICMP <i>Echo Reply</i> packets received.
inTimestamps	The number of ICMP <i>Timestamp Request</i> packets received.
inTimestampReps	The number of ICMP <i>Timestamp Reply</i> packets received.
inAddrMasks	The number of ICMP <i>Address Mask Request</i> packets received.
inAddrMaskReps	The number of ICMP <i>Address Mask Reply</i> packets received.
outMsgs	The number of ICMP packets transmitted.
outErrors	The number of ICMP packets which should have been transmitted but were not.
outDestUnreachs	The number of ICMP <i>Destination Unreachable</i> packets transmitted.
outTimeExcds	The number of ICMP <i>Time Exceeded</i> packets transmitted.
outParamProbs	The number of ICMP <i>Parameter Problem</i> packets transmitted.
outSrcQuenchs	The number of ICMP <i>Source Quench Reply</i> packets transmitted.
outRedirects	The number of ICMP <i>Redirect</i> packets transmitted.

Table 6-17: Parameters displayed in the output of the SHOW IP COUNTER=ICMP command. (Continued)

Parameter	Meaning
outEchos	The number of ICMP <i>Echo Request</i> (ping) packets transmitted.
outEchoReps	The number of ICMP <i>Echo Reply</i> packets transmitted.
outTimestamps	The number of ICMP <i>Timestamp Request</i> packets transmitted.
outTimestampReps	The number of ICMP <i>Timestamp Reply</i> packets transmitted.
outAddrMasks	The number of ICMP <i>Address Mask Request</i> packets transmitted.
outAddrMaskReps	The number of ICMP <i>Address Mask Reply</i> packets transmitted.

Figure 6-18: Example output from the SHOW IP COUNTER=INTERFACE command.

Management Information Block Counters	

Interface Counters	
Interface: eth0	
ifInOctets	249930
ifInUcastPkts	1197
ifInNUcastPkts	0
ifInDiscards	0
ifInErrors	0
ifInUnknownProtos	815
ifOutOctets	2380
ifOutUcastPkts	34
ifOutNUcastPkts	0
ifOutDiscards	0
ifOutErrors	0
Interface: ISDN Basic Rate Interface	
ifInOctets	0
ifInUcastPkts	0
ifInNUcastPkts	0
ifInDiscards	0
ifInErrors	0
ifInUnknownProtos	0
ifOutOctets	0
ifOutUcastPkts	0
ifOutNUcastPkts	0
ifOutDiscards	0
ifOutErrors	0

Table 6-18: Parameters displayed in the output of the SHOW IP COUNTER=INTERFACE command.

Parameter	Meaning
ifInOctets	The number of octets (bytes) received by the interface.
ifInUcastPkts	The number of unicast packets received.
ifInNUcastPkts	The number of multicast packets received.
ifInDiscards	The number of packets discarded.
ifInErrors	The number of packets received with errors.
ifUnknownProtos	The number of packets received but discarded because their protocol is unsupported.
ifOutOctets	The number of bytes transmitted by the interface.
ifOutUcastPkts	The number of unicast packets transmitted.
ifOutNUcastPkts	The number of multicasts transmitted.

Table 6-18: Parameters displayed in the output of the SHOW IP COUNTER=INTERFACE command. (Continued)

Parameter	Meaning
ifOutDiscards	The number of output packets discarded.
ifOutErrors	The number of packets that should have been transmitted but were not transmitted because of errors.

Figure 6-19: Example output from the SHOW IP COUNTER=IP command.

IP counters			
inReceives	1005	outRequests	0
inHdrErrors	0	outDiscards	0
inAddrErrors	0	outNoRoutes	0
inUnknownProtos	0	forwDatagrams	33
inDiscards	0	routingDiscards	0
inDelivers	972		
reasmReqds	0	fragCreates	0
reasmOKs	0	fragOKs	0
reasmFails	0	fragFails	0
IP Gateway Discards			
tinyFragments	0	spoofedPkts	12
invalHdrOption	0	dirBroadcasts	0
saSpoofedPkts	0	saBlockedPkts	0
saEncodeFails	0		

Table 6-19: Parameters displayed in the output of the SHOW IP COUNTER=IP command.

Parameter	Meaning
inReceives	The number of IP packets received by the router.
inHdrErrors	The number of IP packets received which had header errors.
inAddrErrors	The number of IP packets received with address errors.
inUnKnownProtos	The number of IP packets received with unsupported protocols.
inDiscards	The number of IP packets received but discarded due to resource limitations at the IP level.
inDelivers	The number of IP packets received and passed on by the IP software to other modules.
reasmReqds	The number of IP packets received which needed reassembly.
reasmOKs	The number of IP packets successfully reassembled.
reasmFails	The number of reassembly failures.
outRequests	The number of IP packets requested to be transmitted by higher layers.
outDiscards	The number of output IP packets discarded due to resource limitations at the IP level.
outNoRoutes	The number of output IP packets discarded because no route existed to the destination.
forwDatagrams	The number of IP packets forwarded.

Table 6-19: Parameters displayed in the output of the SHOW IP COUNTER=IP command. (Continued)

Parameter	Meaning
routingDiscards	The number of routing entries that were discarded even though they were valid (possibly to free up buffer space).
fragCreates	The number of fragments created.
fragOKs	The number of IP packets which were successfully fragmented.
fragFails	The number of IP packets which needed fragmenting but the IP flags field indicated not to fragment.
tinyFragments	The number of packets discarded because they were part of a tiny fragment attack.
invalHdrOption	The number of packets discarded because they contained an invalid header option.
saSpoofedPkts	The number of packets discarded because they claimed to be from a Security Association partner but were not encoded correctly.
saEncodeFails	The number of packets discarded because the Security Association encoding failed.
spoofedPkts	The number of packets discarded because they were spoofed packets.
dirBroadcasts	The number of packets discarded because directed broadcasts are not allowed.
saBlockedPkts	The number of packets discarded by a Security Association because they originated from addresses that do not belong to the Security Association.

Figure 6-20: Example output from the SHOW IP COUNTER=MULTICAST command.

IP Multicast Counters				
Interface	ifInMultPkts	ifInMultDiscard	ifOutMultPkts	ifOutMultDiscards
eth0	123	2	321	1
eth1	1234	2	12321	3

Table 6-20: Parameters displayed in the output of the SHOW IP COUNTER=MULTICAST command.

Parameter	Meaning
Interface	The name of the interface (e.g. ETH0, PPP0), or "LOCAL" for the local IP interface. If multihoming is enabled (two or more logical interfaces have been assigned to a single layer 2 interface), all interface names will include a hyphen ("-") and the logical interface number.
ifInMultPkts	The number of multicast packets received via the interface.
ifInMultDiscard	The number of multicast packets received via the interface that were discarded.
ifOutMultPkts	The number of multicast packets transmitted via the interface.
ifOutMultDiscards	The number of multicast packets to be transmitted via the interface that were discarded.

Figure 6-21: Example output from the SHOW IP COUNTER=RIP command.

```

IP RIP Counter Summary:
  Input:
    inResponses ..... 6
    inTrigRequests ..... 0
    inTrigResponses ..... 0
    inTrigAcks ..... 0
    inDiscards ..... 0
  Output:
    outResponses ..... 15
    outTrigRequests ..... 0
    outTrigResponses ..... 0
    outTrigAcks ..... 0

```

Table 6-21: Parameters displayed in the output of the SHOW IP COUNTER=RIP command.

Parameter	Meaning
inResponses	The number of RIP response packets received by the router.
inTrigRequests	The number of RIP triggered request packets received by the router.
inTrigResponses	The number of RIP triggered response packets received by the router.
inTrigAcks	The number of RIP triggered acknowledgement packets received by the router.
inDiscards	The number of RIP packets received but discarded due to resource limitations at the IP level.
outResponses	The number of RIP response packets transmitted by the router.
outTrigRequests	The number of RIP triggered request packets transmitted by the router.
outTrigResponses	The number of RIP triggered response packets transmitted by the router.
outTrigAcks	The number of RIP triggered acknowledgement packets transmitted by the router.

Figure 6-22: Example output from the SHOW IP COUNTER=ROUTE command.

Route Counters					
IP address	NextHop	Interface	Metric	Octets rcvd	Octets sent
0.0.0.0	202.36.163.21	eth0	4	984	0
192.168.19.0	202.36.163.21	eth0	3	0	0
192.168.39.0	202.36.163.21	eth0	4	0	0
192.168.42.0	202.36.163.21	eth0	4	0	0
192.168.119.0	202.36.163.21	eth0	4	0	0
192.168.255.0	202.36.163.21	eth0	3	0	0
202.36.163.0	0.0.0.0	eth0	1	81504	1468
202.49.72.0	202.36.163.21	eth0	2	0	0
202.49.74.0	202.36.163.21	eth0	3	0	0
203.97.191.0	202.36.163.21	eth0	3	0	0

Table 6-22: Parameters displayed in the output of the SHOW IP COUNTER=ROUTE command.

Parameter	Meaning
IP address	The IP address of the remote network pointed to by this route. This could be any IP address.
NextHop	The IP address of the next router on the path to the remote network. It will always be an address on one of the router's interfaces.
Interface	The interface over which the next hop is reached. This field is blank if the next hop is over an addressless PPP interface. If multihoming is enabled (two or more logical interfaces have been assigned to a single layer 2 interface), all interface names will include a hyphen ("-") and the logical interface number.
Metric	The 'cost' to reach the remote network. If there are two paths to the same network then the one with the lowest metric is used. If both have the same metric then the first occurrence is taken.
Octets rcvd	The number of octets of data received via this route.
Octets sent	The number of octets of data transmitted via this route.

Figure 6-23: Example output from the SHOW IP COUNTER=UDP command.

UDP counters			
inDatagrams 307	outDatagrams 0
inErrors 0	noPorts 6

Table 6-23: Parameters displayed in the output of the SHOW IP COUNTER=UDP command.

Parameter	Meaning
inDatagrams	The number of UDP packets received by the router.
inErrors	The number of UDP packets dropped because they contained an error at the UDP layer.
outDatagrams	The number of UDP packets transmitted by the router.
noPorts	The number of UDP packets which were dropped because their destination port was not known.

See Also SHOW IP INTERFACE
SHOW IP ROUTE
SHOW SNMP in *Chapter 16, Simple Network Management Protocol (SNMP)*
SHOW TCP

SHOW IP DEBUG

Syntax SHOW IP DEBUG [=1..40]

Description This command displays selected entries from the IP debug queue. The debug queue is enabled using ENABLE IP DEBUG command on page 6-78. Incorrectly formatted IP packet headers are captured for later analysis. The queue can have up to 40 entries, each entry consists of the first 64 bytes from the packet in question.

If no packet number is specified, the command only returns the number of packets in the queue, or that no packets have been found.

The following are possible responses to the SHOW IP DEBUG command:

```
No packets are currently stored in the debug queue.  
<value> packets are currently stored in the debug queue.
```

Some limited analysis of the captured packets is done. The following are possible responses to the SHOW IP DEBUG=*n* command, where *n* is a number between 1 and 40, or the maximum number of packets captured so far.

```
Error = Bad destination or source address  
Error = Packet length exceeds interface mtu  
Error = Bad IP header checksum  
Error = Unknown  
Error = Packet IP header length too short  
Error = Bad IP version
```

An explanation of the possible cause of these problems is beyond the scope of this document.

See Also DISABLE IP DEBUG
ENABLE IP DEBUG
SHOW IP

SHOW IP FILTER

Syntax SHOW IP FILTER[=*filter-number*]

where:

- *filter-number* is a number in the range 0 to 99.

Description This command displays information about filters. If a filter is specified, the patterns in the filter are displayed. If a filter is not specified, the patterns in all filters are displayed (Figure 6-24 on page 6-117, Table 6-24 on page 6-118).

Figure 6-24: Example output from the SHOW IP FILTER command.

IP Filters						
No.	Ent.	Source Port Dest. Port Type	Source Address Dest. Address Act/Pol/Pri	Source Mask Dest. Mask Logging	Session Prot. (C/T)	Size Options Matches
1	1	Any	192.168.163.23	255.255.255.255	Any	No
		Any	192.168.163.39	255.255.255.255	Any	No
		General	Exclude	Off		0
	2	Any	192.168.163.24	255.255.255.255	Any	No
		23	192.168.163.39	255.255.255.255	Any	No
		General	Exclude	Off		0
	3	Any	192.168.163.22	255.255.255.255	Any	No
		23	192.168.163.39	255.255.255.255	Any	No
		General	Exclude	Off		0
	4	Any	192.168.163.21	255.255.255.255	Any	No
		23	192.168.163.39	255.255.255.255	TCP	No
		General	Exclude	Off		0
Requests: 636		Passes: 0		Fails: 636		
2	1	Any	192.168.166.2	255.255.255.255	Any	Yes
		Any	192.168.163.39	255.255.255.255	Any	No
		General	Include	Off		0
	2	Any	192.168.163.21	255.255.255.255	Any	Yes
		23	192.168.163.39	255.255.255.255	TCP	No
		General	Exclude	Off		0
Requests: 0		Passes: 0		Fails: 0		
3	1	2:34	192.168.163.0	255.255.255.0	Start	Yes
		Any	Any	Any	TCP	No
		General	Include	Off		0
Requests: 0		Passes: 0		Fails: 0		

Table 6-24: Parameters displayed in the output of the SHOW IP FILTER command.

Parameter	Meaning
No.	The number of the filter.
Ent.	The entry number in this filter for the pattern.
Source Port	The source IP port for this pattern.
Source Address	The source IP address for this pattern.
Source Mask	The source IP address mask for this pattern.
Session	The type of TCP packet to match, when the Pro field contains "TCP"; one of "START" (the pattern matches TCP packets with the SYN bit set and the ACK bit clear), "ESTABLISHED" (the pattern matches TCP packets with either the SYN bit clear or the ACK bit set), or "ANY" (the pattern matches any TCP packet).
Size	The maximum reassembly size for IP fragments, or "Any" if no maximum size has been set.
Dest. Port	The Destination IP port for this pattern.
Dest. Address	The Destination IP address for this pattern.
Dest. Mask	The Destination IP address mask for this pattern.
Prot. (C/T)	The protocol for this pattern; one of "ANY", "EGP", "ICMP", "OSPF", "TCP" or "UDP". For the ICMP protocol, the ICMP code and type are also listed.
Options	The IP options field for this pattern; one of "Any", "Yes" or "No".
Type	The pattern type; one of "General" or "Specific".
Act/Pol/Pri	The filter action for traffic filters (one of "Exclude" or "Include"), the policy number for policy filters, or the priority of priority filters..
Logging	Whether or not matches to this entry generate log messages to the router's logging facility, and the content of any log messages; one of "Off", "Head", "Dump" or a number in the range 4 to 1600.
Matches	The number of IP packets that have matched this pattern.
Requests	The number of IP packets checked against this filter.
Passes	The number of IP packets included by this filter.
Fails	The number of IP packets excluded by this filter.

See Also ADD IP FILTER
 ADD IP TRUSTED
 DELETE IP FILTER
 DELETE IP TRUSTED
 SET IP FILTER
 SHOW IP TRUSTED

SHOW IP HELPER

Syntax SHOW IP HELPER [COUNTER]

Description This command displays information about the state of broadcast forwarding on the router. If no optional parameters are specified, the current configuration is displayed (Figure 6-25 on page 6-119, Table 6-25 on page 6-119). If the COUNTER parameter is specified, counters for traffic that has been forwarded are displayed (Figure 6-26 on page 6-119, Table 6-26 on page 6-120).

Figure 6-25: Example output from the SHOW IP HELPER command.

```
IP HELPER Configuration

Status : Enabled
-----
Interface : eth0
  UDP port : 137
    Destination(s) ..... 192.168.2.2
  UDP port : 138
    Destination(s) ..... 192.168.2.2
-----
```

Table 6-25: Parameters displayed in the output of the SHOW IP HELPER command.

Parameter	Meaning
Status	The current status of broadcast forwarding; one of "Enabled" or "Disabled".
Interface	The interface on which broadcast UDP packets are received. If multihoming is enabled (two or more logical interfaces have been assigned to a single layer 2 interface), all interface names will include a hyphen ("-") and the logical interface number.
UDP port	The UDP port number(s) to be matched against received UDP broadcast packets. If the port number of a UDP packet matches one on the list then that packet is forwarded to each of the destination IP addresses.
Destination	The destination IP address to which matching broadcast UDP packets will be forwarded.

Figure 6-26: Example output from the SHOW IP HELPER COUNTER command.

```
IP HELPER Counters

-----
Interface : eth0
  InPackets ..... 1
  InNoDestination ..... 0
  Port : 137
    OutPackets ..... 0
  Port : 138
    OutPackets ..... 1
-----
```

Table 6-26: Parameters displayed in the output of the SHOW IP HELPER COUNTER command.

Parameter	Meaning
Interface	The interface on which broadcast UDP packets are received. If multihoming is enabled (two or more logical interfaces have been assigned to a single layer 2 interface), all interface names will include a hyphen ("-") and the logical interface number.
InPackets	The number of broadcast UDP packets received on the interface. Note opening a UDP listen port means that all matching UDP packets received on any interface are processed.
InNoDestination	The number of broadcast UDP packets received on the interface that did not match a requested port.
Port	The UDP port number(s) to be matched against received UDP broadcast packets.
OutPackets	The number of packets forwarded to the specified destinations listed for that UDP port.

Examples To display the current status of broadcast forwarding, use the command:

```
SHOW IP HELPER
```

See Also ADD IP HELPER
DELETE IP HELPER
DISABLE IP HELPER
ENABLE IP HELPER

SHOW IP HOST

Syntax SHOW IP HOST

Description This command displays the IP host name table and the IP address of the nameserver, if defined. (Figure 6-27 on page 6-120, Table 6-27 on page 6-121). A host name can be any arbitrary string and need not be the full domain name. The host name table makes it easier to Telnet to commonly accessed hosts by enabling the user to enter a shorter, easier to remember name for the host rather than the host's full IP address or domain name. When a host name is specified in the TELNET command on page 7-11 of *Chapter 7, Terminal Server*, the entire name will be used to match a name in the host name table. All characters are used in the comparison, including nonalphabetic characters if they are present. The comparison is not case-sensitive.

Figure 6-27: Example output from the SHOW IP HOST command.

IP Address	Host Name
172.16.8.2	ip4
172.16.8.3	Zaphod
172.29.2.8	Admin

Table 6-27: Parameters displayed in the output of the SHOW IP HOST command.

Parameter	Meaning
IP Address	The IP address of an IP host.
Host name	The nickname given to the IP host, which can be used in the TELNET command on page 7-11 of <i>Chapter 7, Terminal Server</i> (e.g. "TELNET zaphod").

See Also ADD IP HOST
 DELETE IP HOST
 SET IP HOST
 SET IP NAMESERVER
 SET IP SECONDARYNAMESERVER

SHOW IP INTERFACE

Syntax SHOW IP INTERFACE [=interface] [COUNTER [=MULTICAST]]

where:

- *interface* is an interface name formed by concatenating a layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number in the range 0 to 15 (e.g. eth0, ppp1-1). If a logical interface is not specified, 0 is assumed (i.e. 'eth0' is equivalent to 'eth0-0').

Description This command displays interface configuration information for the interfaces which have been assigned to the IP module using the ADD IP INTERFACE command on page 6-53. If an interface is specified then information for the specified interface is displayed, otherwise information for all IP interfaces is displayed (Figure 6-28 on page 6-121, Table 6-28 on page 6-122).

The COUNTER parameter displays counters for the specified interface or all interfaces (Figure 6-29 on page 6-123, Table 6-29 on page 6-123).

Figure 6-28: Example output from the SHOW IP INTERFACE command.

Interface	Type	IP Address	Bc	Fr	PArp	Filt	RIP	Met.
Pri. Filt	Pol.Filt	Network Mask	MTU	VJC	GRE	DBcast	Mul.	
-----	-----	-----	-----	-----	-----	-----	-----	-----
LOCAL	-	Not Set	-	n	-	---	--	
---	---	---	---	---	---	---	---	---
eth0	Static	192.168.163.39	1	y	On	---	01	
---	---	255.255.255.0	1500	-	---	No	On	
ppp1	Dynamic	0.0.0.0	1	y	-	---	01	
---	---	255.255.255.255	1500	Off	---	No	On	
ppp2	Inactive	192.168.23.3	1	n	-	---	01	--
---	---	255.255.255.0	1500	Off	---	Yes	Off	
-----	-----	-----	-----	-----	-----	-----	-----	-----

Table 6-28: Parameters displayed in the output of the SHOW IP INTERFACE command.

Parameter	Meaning
Interface	The name of the interface (e.g. ETH0, PPP0), or "LOCAL" for the local IP interface. If multihoming is enabled (two or more logical interfaces have been assigned to a single layer 2 interface), all interface names will include a hyphen ("-") and the logical interface number.
Type	The type of interface; one of "Static", "Dynamic" or "Inactive". A static interface is a permanent interface that is active and in use. A dynamic interface is a non-permanent interface created when a dial-in user initiates a SLIP or PPP connection. The interface will disappear when the user logs off, when the router is restarted or when the IP module is reset with the RESET IP command on page 6-84. An inactive interface is a permanent interface that could not attach to the lower-layer (PPP, ETH, etc) interface for some reason. The interface is not in use but remains configured and will become active if the lower-layer attachment succeeds on the next RESET IP or restart. The most common cause of inactive interfaces is the deletion of the lower-layer interface. Inactive interfaces may be deleted by the manager, but can not be modified.
IP Address	The IP address assigned to this interface. For an interface configured using DHCP, the IP Address field will show the value assigned by DHCP, or 0.0.0.0 if a DHCP reply has not yet been received.
Bc	This is set to 0 if an all '0' broadcast is required and '1' otherwise. It will default to '1'.
PArp	Whether or not this interface supports proxy ARP; one of "On" or "Off". This option is only valid for Ethernet LAN interfaces.
Fr	Whether or not packets larger than the interface MTU will be fragmented, regardless of the setting of the "Do not fragment" bit; one of "Yes" (ignore the "Do not fragment" bit and fragment oversized packets) or "No" (obey the "Do not fragment" bit).
Filt	The number of the traffic filter applied to the interface, or "---" if no traffic filter is assigned.
RIP Met.	The RIP metric associated with transmitting packets over this interface.
Pri. Filt	The number of the priority filter applied to the interface, or "---" if no priority filter is assigned.
Pol.Filt	The number of the policy filter applied to the interface, or "---" if no policy filter is assigned.
Network Mask	The subnet mask assigned to the IP address of this interface. For an interface configured using DHCP, the Network Mask field will show the value assigned by DHCP, or 0.0.0.0 if a DHCP reply has not yet been received.
MTU	The maximum packet size that can be transmitted over this interface.
VJC	Whether or not Van Jacobson's header compression is active on the interface; one of "On" or "Off". This option is only valid for PPP interfaces.
GRE	The number of the GRE entity associated with the interface, or "---" if no GRE entity is assigned.
DBcast	Whether or not network and subnet broadcasts are forwarded to the network attached to the interface; one of "Yes" or "No".

Table 6-28: Parameters displayed in the output of the SHOW IP INTERFACE command. (Continued)

Parameter	Meaning
Mul.	The way multicast packets are handled on the interface; one of "On" (multicast packets are sent and received), "Rec" (multicast packets are received but not sent), "Snd" (multicast packets are sent but not received) or "Off" (multicast packets are neither sent nor received).

Figure 6-29: Example output from the SHOW IP INTERFACE COUNTER command.

IP Interface Counters				
Interface	ifInPkts	ifInBcastPkts	ifInUcastPkts	ifInDiscards
Type	ifOutPkts	ifOutBcastPkts	ifOutUcastPkts	ifOutDiscards
eth0	23531	23224	307	0
Static	230	0	230	0
eth1	0	0	0	0
Static	63289	63289	0	0
ppp0	0	0	0	0
Static	0	0	0	0

Table 6-29: Parameters displayed in the output of the SHOW IP INTERFACE COUNTER command.

Parameter	Meaning
Interface	The name of the interface (e.g. ETH0, PPP0), or "LOCAL" for the local IP interface. If multihoming is enabled (two or more logical interfaces have been assigned to a single layer 2 interface), all interface names will include a hyphen ("-") and the logical interface number.
Type	The type of interface; one of "Static", "Dynamic" or "Inactive". A static interface is a permanent interface that is active and in use. A dynamic interface is a non-permanent interface created when a dial-in user initiates a SLIP or PPP connection. The interface will disappear when the user logs off, when the router is restarted or when the IP module is reset with the RESET IP command on page 6-84. An inactive interface is a permanent interface that could not attach to the lower-layer (PPP, ETH, etc) interface for some reason. The interface is not in use but remains configured and will become active if the lower-layer attachment succeeds on the next RESET IP or restart. The most common cause of inactive interfaces is the deletion of the lower-layer interface. Inactive interfaces may be deleted by the manager, but can not be modified.
ifInPkts	The number of packets received via the interface.
ifOutPkts	The number of packets transmitted via the interface.
ifInBcastPkts	The number of multicast packets received via the interface.
ifOutBcastPkts	The number of multicast packets transmitted via the interface.

Table 6-29: Parameters displayed in the output of the SHOW IP INTERFACE COUNTER command. (Continued)

Parameter	Meaning
ifInUcastPkts	The number of unicast packets received via the interface.
ifOutUcastPkts	The number of unicast packets transmitted via the interface.
ifInDiscards	The number of packets received via the interface that were discarded.
ifOutDiscards	The number of packets to be transmitted via the interface that were discarded.

See Also ADD IP INTERFACE
DELETE IP INTERFACE
DISABLE IP INTERFACE
ENABLE IP INTERFACE
RESET IP INTERFACE
SET IP INTERFACE
SHOW IP COUNTER

SHOW IP POOL

Syntax SHOW IP POOL [=*pool-name*] [IP=*ipadd*[-*ipadd*]] [SUMMARY]

where:

- *pool-name* is a character string, 1 to 15 characters in length. Valid characters are any printable characters. If *pool-name* contains spaces, it must be enclosed in double quotes.
- *ipadd* is an IP address in dotted decimal notation.

Description This command displays information about a single IP address pool or all IP address pools.

The POOL parameter specifies the name of the IP address pool to display. The specified pool must already exist. If a value is not specified, information for all defined IP pools are displayed (Figure 6-30 on page 6-125, Table 6-30 on page 6-125).

The IP parameter limits the display to a specific IP address or range of IP addresses from the pool.

If SUMMARY is specified, only summary information about the specified IP address pool(s) is displayed (Figure 6-31 on page 6-125, Table 6-30 on page 6-125).

Figure 6-30: Example output from the SHOW IP POOL command.

```

IP Pool
-----
Pool Name: dialin ( 192.168.1.1 - 192.168.1.8 )
Number of requests ..... 102
Request successes ..... 101
Request failures ..... 1
Number in use ..... 5
IP Address Interface Status Start Time End time
192.168.1.1 PPP0 inuse 24-Jun-1999 15:21:58
192.168.1.2 PPP1 free 24-Jun-1999 10:02:04 24-Jun-1999 16:23:50
192.168.1.3 PPP2 inuse 24-Jun-1999 15:32:17
192.168.1.4 PPP3 inuse 24-Jun-1999 15:36:01
192.168.1.5 PPP4 inuse 24-Jun-1999 15:37:46
192.168.1.6 PPP5 inuse 24-Jun-1999 15:51:06
192.168.1.7 PPP6 free 24-Jun-1999 15:59:51 24-Jun-1999 16:03:11
192.168.1.8 free never used
-----

```

Table 6-30: Parameters displayed in the output of the SHOW IP POOL command.

Parameter	Meaning
Pool Name	The name of the IP address pool and the IP addresses assigned to the pool.
Number of requests	The total number of requests to allocate an IP address from the specified pool.
Request successes	The number of successful requests to allocate an IP address from the specified pool.
Request failures	The number of failed requests to allocate an IP address from the specified pool.
Number in use	The number of IP addresses currently in use for the specified pool.
IP Address	An IP address in the specified pool.
Interface	The interface that last requested the IP address.
Status	The status of the IP address; one of "inuse" or "free".
Start Time	The date and time the IP address was allocated from the pool.
End Time	The data and time the IP address was released back to the pool.

Figure 6-31: Example output from the SHOW IP POOL SUMMARY command.

```

IP Pool
-----
Pool Name: dialin ( 192.168.1.1 - 192.168.1.16 )
Number of requests ..... 102
Request successes ..... 101
Request failures ..... 1
Number in use ..... 5
-----

```

Examples To display detailed information about the IP address pool named "dialin", use the command:

```
SHOW IP POOL=dialin
```

See Also CREATE IP POOL
DESTROY IP POOL

SHOW IP RIP

Syntax SHOW IP RIP [INTERFACE=*interface*] [CIRCUIT=*miox-circuit*]
[IP=*ipadd*]

where:

- *interface* is an interface name formed by concatenating a layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number in the range 0 to 15 (e.g. eth0, ppp1-1). If a logical interface is not specified, 0 is assumed (i.e. 'eth0' is equivalent to 'eth0-0').
- *miox-circuit* is the name of a MIOX circuit defined for an X.25 interface, 1 to 15 characters in length. The name is not case-sensitive.
- *ipadd* is an IP address in dotted decimal notation.

Description This command displays information about the RIP configuration for IP (Figure 6-32 on page 6-126, Table 6-31 on page 6-126). The INTERFACE, CIRCUIT and IP parameters can be used to restrict the display to RIP neighbours on specific interfaces, MIOX circuits, or with specific IP addresses.

Figure 6-32: Example output from the SHOW IP RIP command.

Interface	Circuit	IP Address	Send	Receive	Demand	Auth	Password
eth0	-	-	COMP	BOTH	NO	NO	
ppp0	-	172.16.249.34	RIP1	RIP2	YES	PASS	*****
ppp1		172.16.250.2	RIP2	NONE	YES	PASS	NOT SET

Table 6-31: Parameters displayed in the output of the SHOW IP RIP command.

Parameter	Meaning
Interface	The interface via which RIP packets are exchanged with the RIP neighbour. If multihoming is enabled (two or more logical interfaces have been assigned to a single layer 2 interface), all interface names will include a hyphen ("-") and the logical interface number.
Circuit	The circuit name if this is an X.25 interface.
IP Address	The IP address of the RIP neighbour.
Send	The type of RIP packets to send; one of "NONE", "RIP1", "RIP2" or "COMP".
Receive	The type of RIP packets to receive; one of "NONE", "RIP1", "RIP2" or "BOTH".
Demand	Whether or not the demand RIP procedures are to be used; one of "YES" or "NO".
Auth	The type of authentication to use with the RIP neighbour; one of "NONE", "PASS", or "MD5".

Table 6-31: Parameters displayed in the output of the SHOW IP RIP command.

Parameter	Meaning
Password	If the authentication type is PASSWORD or MD5, "*****" if a password is set or "NOT SET" if there is no password.

Examples To show the RIP configuration for the Ethernet 0 interface, use the command:

```
SHOW IP RIP INTERFACE=eth0
```

See Also ADD IP RIP
DELETE IP RIP
SET IP RIP
SHOW IP
SHOW IP COUNTER

SHOW IP RIPTIMER

Syntax SHOW IP RIPTIMER

Description This command displays the current settings of the global RIP timers (Figure 6-33 on page 6-127, Table 6-32 on page 6-127).

Figure 6-33: Example output from the SHOW IP RIPTIMER command.

IP RIP timers		
Timer name	Default	Current

Update	30	5
Invalid	180	15
Holddown	120	60
Flush	300	75

Table 6-32: Parameters displayed in the output of the SHOW IP RIPTIMER command.

Parameter	Meaning
Timer name	The name of the timer.
Default	The default value (in seconds) for the timer.
Current	The current value (in seconds) for the timer.
Update	The time interval (in seconds) between RIP updates for all interfaces not using RIP on demand.
Invalid	The time interval (in seconds) after which the router will deem a route to be invalid if no update has been received for the route.
Holddown	The time interval (in seconds), after a route has become invalid, during which the router will ignore updates for the route which would normally make the route valid again.

Table 6-32: Parameters displayed in the output of the SHOW IP RIPTIMER command. (Continued)

Parameter	Meaning
Flush	The time interval (in seconds), from the last update of a route, until the route is flushed from the route table.

Examples To display the current settings of the global RIP timers, use the command:

```
SHOW IP RIPTIMER
```

See Also SET IP RIPTIMER

SHOW IP RIP COUNTER

Syntax `SHOW IP RIP COUNTER [= {DETAIL|SUMMARY}]`
`[INTERFACE=interface] [CIRCUIT=miox-circuit] [IP=ipadd]`

where:

- *interface* is an interface name formed by concatenating a layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number in the range 0 to 15 (e.g. eth0, ppp1-1). If a logical interface is not specified, 0 is assumed (i.e. 'eth0' is equivalent to 'eth0-0').
- *miox-circuit* is the name of a MIOX circuit defined for an X.25 interface, 1 to 15 characters in length. The name is not case-sensitive.
- *ipadd* is an IP address in dotted decimal notation.

Description This command displays counters for RIP (Figure 6-34 on page 6-129, Table 6-33 on page 6-129).

The COUNTER parameter specifies whether to display summary or detailed information. If DETAIL is specified, counters for each RIP neighbour and total counts for all RIP neighbours are displayed. Otherwise, just the total counts for all RIP neighbours are displayed.

The INTERFACE, CIRCUIT and IP parameters can be used to restrict the display to RIP neighbours on specific interfaces, MIOX circuits, or with specific IP addresses.

Figure 6-34: Example output from the SHOW IP RIP COUNTER=DETAIL command.

IP RIP Counters:	
Interface: eth0	
Input:	Output:
inResponses 2568	outResponses 2567
inTrigRequests 0	outTrigRequests 0
inTrigResponses 0	outTrigResponses 0
inTrigAcks 0	outTrigAcks 0
inDiscards 0	
IP RIP Counter Summary:	
Input:	Output:
inResponses 2568	outResponses 2567
inTrigRequests 0	outTrigRequests 0
inTrigResponses 0	outTrigResponses 0
inTrigAcks 0	outTrigAcks 0
inDiscards 0	

Table 6-33: Parameters displayed in the output of the SHOW IP RIP COUNTER command.

Parameter	Meaning
Interface	The interface of the RIP neighbour. If multihoming is enabled (two or more logical interfaces have been assigned to a single layer 2 interface), all interface names will include a hyphen ("-") and the logical interface number.
Circuit	The circuit name if this is an X.25 interface.
IP Address	The IP address of the RIP neighbour.
inResponses	The number of response packets received.
inTrigRequests	The number of triggered request packets received.
inTrigResponses	The number of triggered response packets received.
inTrigAcks	The number of triggered acknowledge packets received.
inDiscards	The number of packets discarded. Packets may be discarded due to authentication failure, packets received when receive is disabled or mismatched sequence number of a triggered acknowledgement.
outResponses	The number of response packets transmitted.
outTrigRequests	The number of triggered request packets transmitted.
outTrigResponses	The number of triggered response packets transmitted.
outTrigAcks	The number of triggered acknowledge packets transmitted.

See Also **SHOW IP COUNTER**
SHOW IP RIP

SHOW IP ROUTE

Syntax `SHOW IP ROUTE [=ipadd] [{GENERAL|CACHE|COUNT}]`

where:

- *ipadd* is an IP address in dotted decimal notation.

Description This command displays information about the IP route table. If no optional parameters are specified, the contents of the route table is displayed (Figure 6-35 on page 6-131, Table 6-34 on page 6-131). If ROUTE is specified with an IP address that does not contain the wildcard character ("*"), the display lists all routes which can be used to reach the specified destination address, including the default route 0.0.0.0. If ROUTE is specified with an IP address that ends with the wildcard character ("*"), the display lists all routes beginning with the specified address. The wildcard character may only be used to replace a complete number in the address, not part of a number. For example, 192.168.*.* is valid and will display all routes in the route table that start with 192.168, but 192.168.12*.* is not valid.

If GENERAL is specified, summary information is displayed (Figure 6-36 on page 6-132, Table 6-35 on page 6-132).

If CACHE is specified, the contents of the route cache is displayed (Figure 6-37 on page 6-132, Table 6-36 on page 6-132). If ROUTE is also specified with an IP address, only those routes in the route cache that were used to forward packets to the destination specified by the IP address are displayed.

If COUNT is specified, summary information about the numbers of octets received and transmitted via each route is displayed (Figure 6-38 on page 6-133, Table 6-37 on page 6-133).

Figure 6-35: Example output from the SHOW IP ROUTE command.

IP Routes					
Destination Circ.	Mask Type	Policy	NextHop Protocol	Interface Metrics	Age Preference
0.0.0.0	0.0.0.0		202.36.163.21	eth0	1
-	remote	0	rip	5	100
192.168.69.0	255.255.255.0		202.36.163.35	eth0	0
-	remote	0	rip	2	100
192.168.201.0	255.255.255.0		202.36.163.21	eth0	1
-	remote	0	rip	5	100
192.168.202.0	255.255.255.0		202.36.163.21	eth0	1
-	remote	0	rip	6	100
192.168.203.0	255.255.255.0		202.36.163.21	eth0	1
-	remote	0	rip	5	100
192.168.204.0	255.255.255.0		202.36.163.21	eth0	1
-	remote	0	rip	3	100
192.168.206.0	255.255.255.0		202.36.163.21	eth0	1
-	remote	0	rip	4	100
202.36.163.0	255.255.255.192		0.0.0.0	eth0	5699
-	direct	0	interface	1	0
202.49.72.0	255.255.255.0		202.36.163.21	eth0	1
-	remote	0	rip	2	100
202.49.73.0	255.255.255.0		202.36.163.21	eth0	1
-	remote	0	rip	3	100
202.49.75.0	255.255.255.0		202.36.163.21	eth0	1
-	remote	0	rip	3	100

Table 6-34: Parameters displayed in the output of the SHOW IP ROUTE command.

Parameter	Meaning
Destination	The IP address of the destination network.
Mask	The subnet mask for the route.
NextHop	The IP address of the next router on the route to the destination, or the <i>ifindex</i> of an addressless PPP interfaces in dotted decimal notation.
Interface	The interface via which the destination network can be reached. If multihoming is enabled (two or more logical interfaces have been assigned to a single layer 2 interface), all interface names will include a hyphen ("-") and the logical interface number.
Age	The time in seconds that the route has been known.
Circ.	The MIOX circuit via which the destination network can be reached, for X.25 interfaces.
Type	The type of route; one of "remote", "direct" or "other".
Policy	The policy number of this route.
Protocol	The protocol used to determine the route; one of "static" or "rip".
Metrics	The routing metric (cost) to reach the destination network.
Preference	The routing preference value. Routes with a high preference (low value) will be used ahead of routes with a low preference (high value).

Figure 6-36: Example output from the SHOW IP ROUTE GENERAL command.

```

IP Route General Information
-----
Number of routes ..... 12
Cache size ..... 1024
Source route byte counting ..... no
Route debugging ..... no
Multipath routing ..... yes

```

Table 6-35: Parameters displayed in the output of the SHOW IP ROUTE GENERAL command.

Parameter	Meaning
Number of routes	The number of routes in the route table.
Cache size	The size of the route cache, in bytes.
Source route byte counting	Whether or not source route byte counting is enabled; one of "yes" or "no".
Route debugging	Whether or not route debugging is enabled; one of "yes" or "no".
Multipath routing	Whether or not multipath route is enabled; one of "yes" or "no".

Figure 6-37: Example output from the SHOW IP ROUTE CACHE command.

```

IP Route Cache
-----
Destination      Route           Route mask      Nexthop          Interface
-----
202.36.163.4     202.36.163.0   255.255.255.192 0.0.0.0          eth0
202.36.163.5     202.36.163.0   255.255.255.192 0.0.0.0          eth0
202.36.163.6     202.36.163.0   255.255.255.192 0.0.0.0          eth0
202.36.163.11    202.36.163.0   255.255.255.192 0.0.0.0          eth0
202.36.163.21    202.36.163.0   255.255.255.192 0.0.0.0          eth0
202.36.163.31    202.36.163.0   255.255.255.192 0.0.0.0          eth0
202.36.163.36    202.36.163.0   255.255.255.192 0.0.0.0          eth0
202.36.163.51    202.36.163.0   255.255.255.192 0.0.0.0          eth0
202.36.163.61    202.36.163.0   255.255.255.192 0.0.0.0          eth0
202.36.163.5     202.36.163.0   255.255.255.192 0.0.0.0          eth0
             hits:      875             misses:      11
-----

```

Table 6-36: Parameters displayed in the output of the SHOW IP ROUTE CACHE command.

Parameter	Meaning
Destination	The destination IP address.
Route	The route used to forward packets to the destination IP address.
Route mask	The network mask for the route.
NextHop	The next hop on the route.

Table 6-36: Parameters displayed in the output of the SHOW IP ROUTE CACHE command. (Continued)

Parameter	Meaning
Interface	The interface via which the destination network can be reached. If multihoming is enabled (two or more logical interfaces have been assigned to a single layer 2 interface), all interface names will include a hyphen ("-") and the logical interface number.

Figure 6-38: Example output from the SHOW IP ROUTE COUNT command.

Route Counters					
IP address	NextHop	Interface	Metric	Octets rcvd	Octets sent
192.168.1.0	202.36.163.21	eth1	1	0	0
192.168.1.0	202.36.163.21	eth1	1	0	0
192.168.1.64	202.36.163.21	eth1	1	0	0
192.168.1.128	202.36.163.21	eth1	1	0	0
192.168.1.192	202.36.163.21	eth1	1	0	0
192.168.1.208	202.36.163.21	eth1	1	0	0

Table 6-37: Parameters displayed in the output of the SHOW IP ROUTE COUNT command.

Parameter	Meaning
IP address	The IP address of the destination to which packets were transmitted using this route.
NextHop	The IP address of the next router on the route to the destination.
Interface	The interface via which the destination network can be reached. If multihoming is enabled (two or more logical interfaces have been assigned to a single layer 2 interface), all interface names will include a hyphen ("-") and the logical interface number.
Metric	The routing metric (cost) to reach the destination network.
Octets rcvd	The number of octets received via this route.
Octets sent	The number of octets transmitted via this route.

See Also ADD IP ROUTE
 DELETE IP ROUTE
 SET IP ROUTE

SHOW IP ROUTE FILTER

Syntax SHOW IP ROUTE FILTER

Description This command displays information about configured IP route filters (Figure 6-39 on page 6-134, Table 6-38 on page 6-134).

Figure 6-39: Example output from the SHOW IP ROUTE FILTER command.

IP Route Filters					
Ent.	IP Address Protocol	Mask Direction	Nexthop Interface	Policy Action	Matched
1	0.0.0.0 RIP	0.0.0.0 Both	Any -	0 Include	0
Request: 1		Passes: 1		Fails: 0	

Table 6-38: Parameters displayed in the output of the SHOW IP ROUTE FILTER command.

Parameter	Meaning
Ent.	The filter number.
IP Address	The IP address of the network to be filtered.
Mask	The network mask for the network address.
Nexthop	The next hop to which the filter applies.
Policy	The policy or type of service to which the filter applies.
Matched	The number of times this pattern has been matched.
Protocol	The routing protocol to which the filter applies.
Direction	The direction to which the filter applies; one of "Receive", "Send" or "Both".
Interface	The interface to which the filter applies. If multihoming is enabled (two or more logical interfaces have been assigned to a single layer 2 interface), all interface names will include a hyphen ("-") and the logical interface number.
Action	The action, when a route matches the pattern; one of "Include" or "Exclude".

See Also ADD IP ROUTE FILTER
DELETE IP ROUTE FILTER
SET IP ROUTE FILTER

SHOW IP ROUTE TEMPLATE

Syntax `SHOW IP ROUTE TEMPLATE [=name]`

where:

- *name* is a character string, 1 to 31 characters in length. Valid characters are any printable character. If *name* contains spaces it must be enclosed in double quotes. *name* is not case-sensitive.

Description This command displays information about the specified or all IP route templates. If a template is not specified, summary information about all IP route templates is displayed (Figure 6-40 on page 6-135, Table 6-39 on page 6-135). If a template is specified, detailed information about the specified template is displayed (Figure 6-41 on page 6-135, Table 6-40 on page 6-135).

Figure 6-40: Example output from the SHOW IP ROUTE TEMPLATE command.

Template	Interface
-----	-----
branch_office	ppp0
home	ppp0
-----	-----

Table 6-39: Parameters displayed in the output of the SHOW IP ROUTE TEMPLATE command.

Parameter	Meaning
Template	The name of the IP route template.
Interface	The IP interface specified by the IP route template.

Figure 6-41: Example output from the SHOW IP ROUTE TEMPLATE command.

IP route template	branch_office
Interface	ppp0
Next hop	192.168.23.3
Rip metric	DEFAULT (1)
Policy	DEFAULT (0)
Preference	90

Table 6-40: Parameters displayed in the output of the SHOW IP ROUTE TEMPLATE command.

Parameter	Meaning
IP route template	The name of the IP route template.
Interface	The IP interface specified by the IP route template.
Next hop	The next hop specified by the IP route template.
Rip metric	The rip metric specified by the IP route template.
Policy	The policy specified by the IP route template.
Preference	The preference specified by the IP route template.

Examples To display detailed information about the IP route template named "branch_office", use the command:

```
SHOW IP ROUTE TEMPLATE=branch_office
```

See Also ADD IP ROUTE TEMPLATE
DELETE IP ROUTE TEMPLATE
SET IP ROUTE TEMPLATE

SHOW IP TRUSTED

Syntax SHOW IP TRUSTED

Description This command displays the contents of the trusted router table and the state of the enable flag (Figure 6-42 on page 6-136). The trusted router table ensures that the router's routing table is updated only by *trusted* sources of routing information. Other routers will not be filtered, but their routing information will not be used until they are added to the table.

Figure 6-42: Example output from the SHOW IP TRUSTED command.

Host address

172.16.8.33

See Also ADD IP FILTER
ADD IP TRUSTED
DELETE IP FILTER
DELETE IP TRUSTED
SET IP FILTER
SHOW IP FILTER

SHOW IP UDP

Syntax SHOW IP UDP

Description This command displays the state of current UDP sessions (Figure 6-43 on page 6-136, Table 6-41 on page 6-137). UDP *listens* for SNMP packets and RIP. It will also show a connection when the TFTP download is initiated as part of loading new software.

Figure 6-43: Example output from the SHOW IP UDP command.

Local port	Local address	Remote port
-----	-----	-----
00520	0.0.0.0	00000
00161	0.0.0.0	00000
-----	-----	-----

Table 6-41: Parameters displayed in the output of the SHOW IP UDP command.

Parameter	Meaning
Local port	The port number for the UDP connection on this router. See Table 6-9 on page 6-46 for a list of commonly used, assigned UDP port numbers.
Local address	The IP address for the UDP connection on this router.
Remote port	The port number for the UDP connection on the remote host. See Table 6-9 on page 6-46 for a list of commonly used, assigned UDP port numbers.

See Also SHOW IP COUNTER
 SHOW TCP

SHOW PING

Syntax SHOW PING

Description This command displays information about the PING configuration and the results of the current (if any) or previous PING command (Figure 6-44 on page 6-138, Table 6-42 on page 6-138).

Figure 6-44: Example output from the SHOW PING command.

```

Ping Information
-----
Defaults:
Type ..... IP
Source ..... 0.0.0.0
Destination ..... 192.168.2.1
Number of packets ..... 10
Size of packets (bytes) ..... 24
Timeout (seconds) ..... 1
Delay (seconds) ..... 1
Data pattern ..... Not set
Type of service ..... 0
Direct output to screen ..... Yes

Current:
Type ..... IP
Source ..... 0.0.0.0
Destination ..... 192.168.2.1
Number of packets ..... 10
Size of packets (bytes) ..... 24
Timeout (seconds) ..... 1
Delay (seconds) ..... 1
Data pattern ..... 0x00000000
Type of service ..... 0
Direct output to screen ..... Yes

Results:
Ping in progress ..... No
Packets sent ..... 10
Packets received ..... 10
Round trip time minimum (ms) .. 20
Round trip time average (ms) .. 22
Round trip time maximum (ms) .. 40
Last message ..... Finished successfully
-----

```

Table 6-42: Parameters displayed in the output of the SHOW PING command.

Parameter	Meaning
Type	The network protocol type; "IP".
Source	The source IP address used in the ping packet.
Destination	The IP address or host name to ping.
Number of packets	The number of ping packets to send.
Size of packets (bytes)	The number of data pattern bytes to include in the packet.
Timeout (seconds)	The time, in seconds, to wait for a reply before sending the next packet.
Delay (seconds)	The time, in seconds, to wait before sending the next packet.
Data pattern	The data bytes to be used in the data portion of packets transmitted.

Table 6-42: Parameters displayed in the output of the SHOW PING command. (Continued)

Parameter	Meaning
Type of service	The value of the TOS (Type Of Service) field in the IP header of IP ping packets transmitted.
Direct output to screen	Whether or not the output is sent to the terminal; one of "Yes" or "No".
Ping in progress	Whether or not a ping is in progress; one of "Yes" or "No".
Packets sent	The number of packets sent.
Packets received	The number of packets received.
Round trip time minimum (ms)	The quickest round trip time in milliseconds.
Round trip time average (ms)	The average round trip time in milliseconds.
Round trip time maximum (ms)	The slowest round trip time in milliseconds.
Last message	The last message from the PING command on page 6-82.

Examples To display the current ping configuration, use the command:

```
SHOW PING
```

See Also PING
SET PING
STOP PING

SHOW TCP

Syntax SHOW TCP [=*tcb*]

where:

- *tcb* is the index of a TCP connection in the TCP connection table.

Description This command displays the state of current TCP connections. If a TCP connection is specified, detailed information about the specified TCP connection is displayed (Figure 6-45 on page 6-140, Table 6-43 on page 6-140). If a TCP connection is not specified, the TCP portion of the MIB-II MIB and summary information about all current TCP connections is displayed (Figure 6-46 on page 6-141, Table 6-44 on page 6-142).

This command is useful to show if any Telnet or other TCP sessions are currently active. Port 23 is typically reserved for Telnet. When a Telnet session is active, the IP address of the source and destination will allow the particular session to be identified.

In Figure 6-46 on page 6-141, the two lines:

```
established 00127 172.16.253.2 00023 172.16.8.5
established 00133 172.16.253.2 00023 172.16.8.5
```

indicate *locally* sourced Telnet sessions. These are from the asynchronous ports attached to the router. The next two lines:

```
established 00023 172.16.40.254 00002 172.16.248.51
```

established 00023 172.16.40.254 02123 172.16.9.190

indicate *remotely* sourced Telnet sessions. See Table 6-9 on page 6-46 for a list of commonly used, assigned TCP port numbers.

Figure 6-45: Example output from the SHOW TCP command for a specified TCP connection.

```
TCB: 05 Local: 192.168.35.45,00023 Remote: 192.168.35.61,01032
State: ESTAB O/P State: IDLE
SND.UNA: 0047376265 SND.NXT: 0047376265 SND.WND: 04096
Last Seq: 0641204304 Last Ack: 0047376265
SendCon: 06022 DataCount: 0000000000
RCV.NXT: 0641204305 RCV.WND: 00000
Round Trip Time
SendSrt: 00218 Deviation: 00013 SendReXmit: 00033
Timers:
Event      Time (cs)
No events in timer queue
Fragment list:
Sequence   Length   End sequence
No fragments in fragment list
```

Table 6-43: Parameters displayed in the output of the SHOW TCP command for a specified TCP connection.

Parameter	Meaning
TCB	The index into the TCP connection table for this connection.
Local	The local IP address and port for the connection. See Table 6-9 on page 6-46 for a list of commonly used, assigned TCP port numbers.
Remote	The Remote IP address and port for the connection. See Table 6-9 on page 6-46 for a list of commonly used, assigned TCP port numbers.
State	The state of the connection; one of "FREE", "CLOSD", "LISTN", "SYNSN", "SYNRC", "ESTAB", "FINW1", "FINW2", "CLOSW", "LSTAK", "CLOSG", "TIMEW" or "DELET".
O/P State	The output queue state; one of "IDLE", "PERST" (remote host has closed its receive window and router is transmitting data one character at a time to aid the process of re-opening the window), "TRANS" (there is data to transmit) or "RETRN" (the router is retransmitting data).
SND.UNA	The sequence number of the last unacknowledged octet transmitted over the connection.
SND.NXT	The sequence number of the next octet to be transmitted over the connection.
SND.WND	The transmit window for the connection.
Last Seq	The packet received from the connection.
Last Ack	The last acknowledgement received from the connection.
SendCon	Internal congestion parameter.
DataCount	Number of data octets transmitted over this connection.

Table 6-43: Parameters displayed in the output of the SHOW TCP command for a specified TCP connection. (Continued)

Parameter	Meaning
RCV.NXT	The next octet expected from the connection.
RCV.WND	The receive window for the connection.
SendSrt, Deviation, SendReXmit	Round trip time parameters used to implement Van Jacobson's retransmit time algorithm.
Event	An event on the timer queue; one of "NONE", "SEND" (transmit data), "PERSIST" (transmit data one character at a time if in PERSIST state), "TRANSMIT" (retransmit data) or "DELETE" (clear TCP connection).
Time (cs)	The time to this event (in centiseconds).
Sequence	The first sequence number of a fragment waiting for defragmentation.
Length	The length of the fragment.
End sequence	The last sequence number of the fragment.

Figure 6-46: Example output from the SHOW TCP command.

```

TCP MIB parameters, counters and connections
-----
RTO Algorithm:                vanj
RTO Min (ms):                0000000500    RTO Max (ms):                0000020000

Maximum connections:          00040

Active Opens:                 00004    Passive Opens:                00005
Attempt Fails:                00000    Established Resets:           00000
Current Established:           00004

In Segs:                      0000000070    In Segs Error:                0000000000
Out Segs:                     0000000104    Out Segs Retran:              0000000000
Out Segs With RST:            0000000000

Connection Table:
Index   State      Local port and address  Remote port and address
-----
00      listen     00023  0.0.0.0                00000  0.0.0.0
01      listen     00515  0.0.0.0                00000  0.0.0.0
02      listen     01998  0.0.0.0                00000  0.0.0.0
03      listen     05025  0.0.0.0                00000  0.0.0.0
04      listen     05026  0.0.0.0                00000  0.0.0.0
05      established 00127  172.16.253.2           00023  172.16.8.5
06      established 00133  172.16.253.2           00023  172.16.8.5
07      established 00023  172.16.40.254          00002  172.16.248.51
08      established 00023  172.16.40.254          02123  172.16.9.190
-----

```

Table 6-44: Parameters displayed in the output of the SHOW TCP command.

Parameter	Meaning
RTO Algorithm	The retransmit time algorithm.
RTO Min (ms), RTO Max (ms)	Retransmit time algorithm parameters (milliseconds)
Maximum connections	The maximum number of TCP connections allowed.
Active Opens	The number of active TCP opens. Active opens initiate connections.
Passive Opens	The number of TCP passive opens. Passive opens are issued to wait for a connection from another host.
Attempt Fails	The number of failed connection attempts.
Established Resets	The number of connections which were established but have since been reset.
Current Established	The number of current connections.
In Segs	The number of segments received.
In Segs Error	The number of segments received with an error.
Out Segs	The number of segments transmitted.
Out Segs Retran	The number of segments which were retransmitted.
Out Segs With RST	The number of segments transmitted with the RST bit set.
Index	The entry number in the table.
State	The state of the session (see Table 6-45 on page 6-142 for a list). These are the names of the various states in the TCP state diagram. For more detailed information, refer to the RFC or a text on TCP/IP.
Local port and address	The router's TCP port number and IP address. See Table 6-9 on page 6-46 for a list of commonly used, assigned UDP port numbers.
Remote Port and address	The TCP port number and IP address of the remote host. See Table 6-9 on page 6-46 for a list of commonly used, assigned UDP port numbers.

Table 6-45: TCP states.

State	Meaning
CLOSED	This is the starting state and should not be present at any time since the server module should immediately go into the LISTEN state.
LISTEN	This is termed a <i>passive open</i> and is entered when the server module is waiting for external connections to be made.
SYNSENT	The server will enter this state when a connection is being initiated from a local session and also when a remotely initiated session is being set up just prior to entering the ESTABLISHED state.
SYNRECEIVED	This state is entered when a SYN packet is received indicating that a remote system is attempting to establish a session.

Table 6-45: TCP states. (Continued)

State	Meaning
ESTABLISHED	This state indicates that a connection has been made and is currently active. Data packets can now flow in both directions.
FINWAIT1	This state indicates the first step of a locally initiated termination of a session. The CLOSEWAIT state indicates a <i>remote</i> station is initiating the termination.
FINWAIT2	This state is also part of the local termination process and is required to ensure that no data in transit is lost.
CLOSEWAIT	This state is entered when the remote entity has sent a FIN packet to terminate this link. The server entity will send an ACK packet.
LASTACK	The ACK packet from above will cause the remote system to send a CLOSE packet and the server will enter this state and send a FIN packet thereby terminating this link.
CLOSING	This state is entered when the established local session has initiated a termination (gone to FINWAIT1) and received a FIN packet from the remote entity indicating that it can now terminate also. This is an alternate path to FINWAIT2.
TIMEWAIT	This state may be entered as part of the termination process while waiting for a remote entity to respond to the final ACK packet. The session is then closed.

See Also SHOW IP COUNTER
SHOW IP UDP

SHOW TRACE

Syntax SHOW TRACE

Description This command displays information about the current trace route configuration and the result of the current or previous trace route operation (Figure 6-47 on page 6-144, Table 6-46 on page 6-144).

Figure 6-47: Example output from the SHOW TRACE command.

```

Trace information
-----
Defaults:
Destination ..... 121.23.5.4
Source ..... 202.36.163.31
Number of packets per hop ..... 3
Timeout (seconds) ..... 1
Type of service ..... 8
Port ..... 33434
Minimum time to live ..... 1
Maximum time to live ..... 20
Addresses only output ..... Yes
Direct output to screen ..... Yes

Current:
Destination ..... 206.123.21.3
Source ..... 202.36.163.31
Number of packets per hop ..... 3
Timeout (seconds) ..... 1
Type of service ..... 8
Port ..... 33434
Minimum time to live ..... 1
Maximum time to live ..... 12
Addresses only output ..... Yes
Direct output to screen ..... Yes

Results:
Trace route in progress ..... No

  1. 202.36.163.21      20      20      20 (ms)
  2. 202.49.72.62       0       0       0 (ms)
  3. 203.97.191.65      0       0       0 (ms)
  4. 203.97.191.22     80      93     100 (ms)
  5. 140.200.128.2      40      46      60 (ms)
  6. 131.119.17.205    460     473     480 (ms)
  7. 131.119.0.129     540     553     560 (ms)
  8. 4.0.1.90          800     800     800 (ms)
  9. 4.0.1.14          440     440     440 (ms)
 10. 198.32.136.39     480     480     480 (ms)
 11. 140.223.9.21      520     520     520 (ms)
 12. 140.223.9.18      560     560     560 (ms)

Last message ..... Target unreachable
-----

```

Table 6-46: Parameters displayed in the output of the SHOW TRACE command.

Parameter	Meaning
Destination	The destination IP address or host name.
Source	The source IP address to use in the packets transmitted.
Number of packets per hop	The number of packets to transmit to each hop on the route.
Timeout	The time, in seconds, to wait for a reply before sending the next packet.
Type of service	The value of the TOS field in the IP header of packets transmitted.

Table 6-46: Parameters displayed in the output of the SHOW TRACE command.

Parameter	Meaning
Port	The destination UDP port number.
Minimum time to live	The minimum TTL (Time To Live), used to skip some hops at the start of the route.
Maximum time to live	The maximum hops to which packets will be transmitted.
Addresses only output	Whether or not address to name translation is performed for the output; one of "Yes" or "No".
Direct output to screen	Whether or not the output is sent to the terminal; one of "Yes" or "No".
Trace route in progress	Whether or not a trace route is in progress; one of "Yes" or "No".
1- <i>n</i>	The hop number, IP address, and the maximum, minimum and average round trip time, in milliseconds, to each hop on the route.
Last message	The last message from the PING command on page 6-82.

Examples To show the current trace route configuration, use the command:

```
SHOW TRACE
```

See Also SET TRACE
STOP TRACE
TRACE

STOP PING

Syntax STOP PING

Description This command stops a ping in progress.

Examples To stop a ping in progress, use the command:

```
STOP PING
```

See Also PING
SET PING
SHOW PING

STOP TRACE

Syntax STOP TRACE

Description This command stops a trace route in progress.

Examples To stop a trace route that is in progress, use the command:

```
STOP TRACE
```

See Also SHOW TRACE
STOP TRACE
TRACE

TRACE

Syntax TRACE [[IPADDRESS=] *ipadd*] [MAXTTL=*number*] [MINTTL=*number*]
[NUMBER=*number*] [PORT=*port-number*] [SCREENOUTPUT={YES |
NO}] [SOURCE=*ipadd*] [TIMEOUT=*number*] [TOS=*number*]

where:

- *ipadd* is an IP address in dotted decimal notation or a host name from the host name table.
- *number* is a decimal number.
- *port-number* is an IP port number.

Description This command performs a trace route. The parameters on this command override the defaults set with the SET TRACE command on page 6-104.

If there is no default destination and a destination is not specified on the TRACE command on page 6-146 then a trace is not performed and an error message is displayed.

The IPADDRESS parameter specifies the destination IP address. The command will trace the route to this IP address.

The MAXTTL parameter specifies the maximum value for the TTL (*Time To Live*) field in the IP packet, and is used to limit the trace route to a maximum number of hops. If MAXTTL is not specified then the current default is used.

The MINTTL parameter specifies the initial value of the TTL (*Time To Live*) field in the IP packet, and can be used to skip some hops at the start of the route. If MINTTL is not specified the current default is used.

The NUMBER parameter specifies the number of packets to send to each hop. If NUMBER is not specified the current default is used. A maximum of 100 packets may be transmitted.

The PORT parameter specifies the UDP destination port number for the packets being transmitted, and can be used to detect whether or not there is an IP device listening on the specified port. If an IP device is listening on the port, the ICMP “unreachable” message which trace route depends on will not be returned.

The SCREENOUTPUT parameter specifies whether or not the output is sent to the terminal. If SCREENOUTPUT is not specified then the current default is used.

The SOURCE parameter specifies the IP address to use as a source address in the packets. If SOURCE is not specified then the IP address of the interface from which the packets are transmitted is used.

The TIMEOUT parameter specifies the length of time to wait for a response before sending packets to the next hop. If TIMEOUT is not specified then the

current default is used. If ICMP “unreachable” messages are received within the timeout period, then packets are transmitted to the next hop immediately.

The TOS parameter specifies the value of the TOS (*Type Of Service*) field in the IP header of the packets being transmitted, as a decimal number in the range 0 to 255. If TOS is not specified then the current default is used.

See Also SET TRACE
SHOW TRACE
STOP TRACE

Chapter 7

Terminal Server

Introduction	7-2
TTY Devices	7-2
Command Line Editing and Recall	7-4
Accessing Telnet Hosts	7-5
Command Reference	7-6
SET TELNET	7-6
SET TTY	7-7
SHOW TTY	7-8
TELNET	7-11

Introduction

This chapter describes the terminal server facilities provided by the router, how to configure virtual terminals and create host nickname tables. See *Chapter 2, Interfaces* for details of how to configure and manage the asynchronous ports on the router.

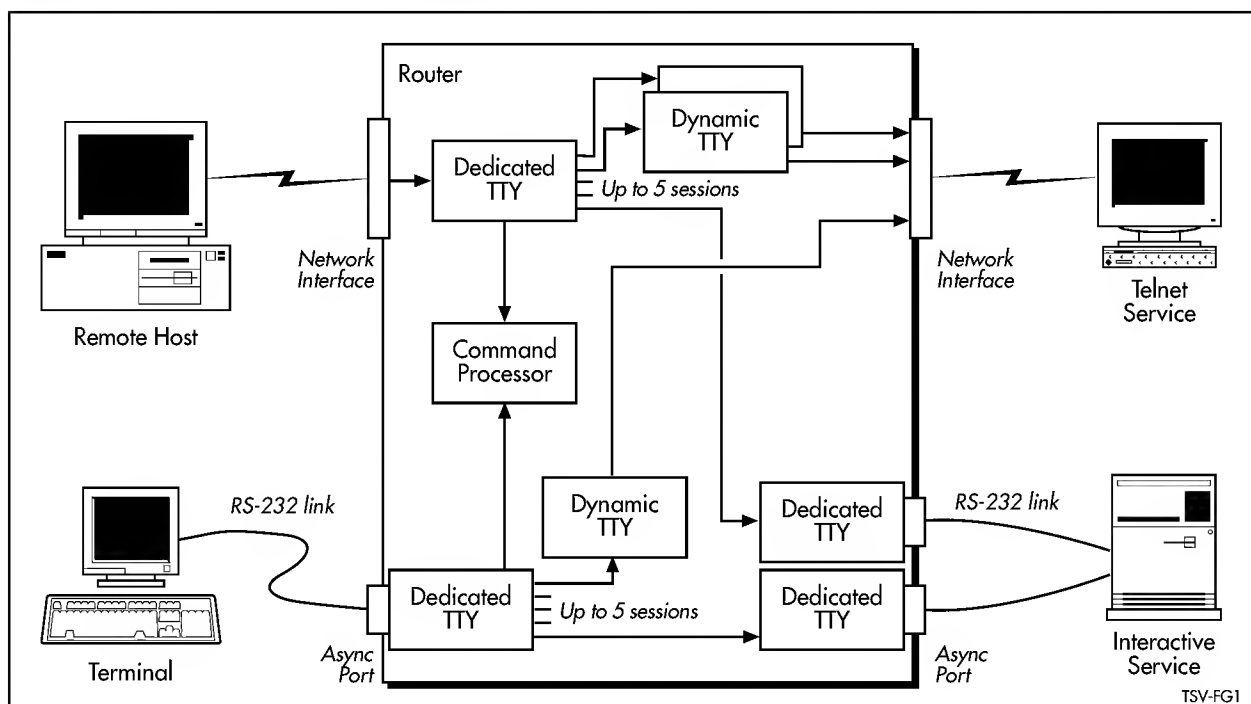
Some of the commands described in this section are available as both MANAGER level and USER level commands. Normally, the commands requiring only USER privilege have a restricted syntax. For example, the SET PORT command, if issued from a port with USER privilege, can only alter the characteristics of the port from which the command was entered. The displays produced by some commands are different for USER and MANAGER privilege. In all cases, the distinction between USER and MANAGER versions will be clearly stated in the command descriptions.

TTY Devices

A TTY device is a *software device* that forms a link between a terminal device (a terminal connected to an asynchronous port or a Telnet connection), and the router's command processor or the Telnet services provided by the router. TTY devices are used to support Telnet connections to the router and multiple terminal sessions from a single asynchronous port or Telnet connection. The term TTY is derived from the UNIX operating system where TTY is an abbreviation for *TeleTYpe*, the terminal I/O handling software layer in UNIX. TTY devices are also called *virtual terminals*.

Each asynchronous port has associated with it a dedicated TTY device which provides access to the router's command prompt, and Telnet services (Figure 7-1 on page 7-2). When a terminal session is initiated to a Telnet service (using the TELNET command on page 7-11), a dynamic TTY device is created for the Telnet service and linked to the dedicated TTY device.

Figure 7-1: TTY devices provide an interface between terminals and Telnet connections, the router's command processor, and interactive and Telnet services provided by the router.



TSV-FG1

Some configuration parameters of a terminal connection to an asynchronous port (such as the baud rate) apply only to the physical port, whereas other parameters (such as the prompt) apply to the dedicated TTY device (Table 7-1 on page 7-3). The SET PORT command on page 2-19 of *Chapter 2, Interfaces* can be used to set the configuration of both the physical port and the dedicated TTY device.

Table 7-1: Configuration parameters for TTY devices.

Parameter	Meaning
HISTORY	Sets the number of commands saved for command line recall.
PAGE	Sets the number of lines of output displayed on the terminal before the router pauses and waits for the user to press a key to continue.
PROMPT	Sets the router prompt to a string, the default prompt, or disables the prompt.
TYPE	Sets the type of the terminal to "VT100" or "DUMB". The DUMB terminal type is used for printing or terminals that do not support VT100 escape sequences.

The SET PORT command on page 2-19 of *Chapter 2, Interfaces*, when executed from a terminal attached to an asynchronous port, displays the configuration of the asynchronous port, followed by the configuration of the dedicated TTY device. The command:

```
SHOW TTY
```

can be used to view just the configuration of the dedicated TTY device.

A Telnet connection (created when a user Telnets to the router) also has an associated dedicated TTY device (Figure 7-1 on page 7-2). The TTY device is temporary, however, and exists only while the Telnet connection is active. When a terminal session is initiated to a Telnet service (using the TELNET command on page 7-11), a dynamic TTY device is created for the Telnet service and linked to the dedicated TTY device.

The SET PORT command on page 2-19 of *Chapter 2, Interfaces* can be used to set the configuration of the dedicated TTY device (Table 7-1 on page 7-3). Physical port characteristics may not be set since, by nature, a Telnet connection is not associated with an asynchronous port.

The SET PORT command on page 2-19 of *Chapter 2, Interfaces*, when executed from a Telnet connection, displays only the configuration of the dedicated TTY device, and is effectively equivalent to the SHOW TTY command on page 7-8.

From either a terminal attached to an asynchronous port or a Telnet connection, the command:

```
SHOW TTY [=tty-number]
```

can be used to display the configuration of any TTY device, including both dedicated and dynamically created TTY devices. The command:

```
SHOW TTY [=tty-number] SUMMARY
```

displays a one-line summary for the specified TTY device. If ALL is specified as the TTY device, a list of all existing dedicated and dynamic TTY devices is displayed.

The command:

```
SET TTY [HISTORY=0..99] [PAGE=4..99] [PROMPT={prompt|DEFAULT|
OFF}] [TYPE={DUMB|VT100}]
```

sets the default configuration for dynamically created TTY devices dedicated to Telnet connections. The default configuration for TTY devices dedicated to the asynchronous ports is set with the SET PORT command on page 2-19 of *Chapter 2, Interfaces*. All defaults are retained through router reboots.

Command Line Editing and Recall

The router supports command line editing and recall. The functions available are:

- Move the cursor backwards and forwards in the command line, using the cursor keys.
- Move the cursor to either end of the command line with a single keystroke.
- Insert and delete characters.
- Clear the command line.
- Toggle between insert and overstrike editing modes.
- Recall, edit and execute previous commands.
- Move backwards and forwards through a history of previous commands.
- Display a command history and select a command from the list.
- Clear the command history.
- Recall the most recent command matching a partially entered command.

Table 7-2 on page 7-4 lists the functions and the terminal keys or key combinations used to access these functions.

Table 7-2: Command line editing functions and keystrokes.

Function	VT100 Terminal	Dumb terminal
Move cursor within command line	←, →	<i>Not available</i>
Delete character to left of cursor	[Delete] or [Backspace]	[Delete] or [Backspace]
Toggle between insert/overstrike	[Ctrl/O]	<i>Not available</i>
Clear command line	[Ctrl/U]	[Ctrl/U]
Recall previous command	↑ or [Ctrl/B]	[Ctrl/B]
Recall next command	↓ or [Ctrl/F]	[Ctrl/F]
Display command history	[Ctrl/C] or SHOW PORT HISTORY	[Ctrl/C] or SHOW PORT HISTORY
Clear command history	RESET PORT HISTORY	RESET PORT HISTORY
Recall matching command	[Tab] or [Ctrl/I]	[Tab] or [Ctrl/I]
Terminate Telnet session	[Ctrl/D]	[Ctrl/D]

The router assumes that the width of the terminal screen is 80 characters, and performs command line wrapping at the 80th column regardless of the setting of the terminal. The cursor does not need to be at the end of the line for the

command to be executed. The default editing mode is insert mode. Characters are inserted at the cursor position and any characters to the right of the cursor are pushed to the right to make room. In overstrike mode, characters are inserted at the cursor position and replace any existing characters.

For VT100-compatible terminals, recalled commands are all displayed on the same line, but for dumb terminals recalled commands are displayed on succeeding lines. The maximum number of commands stored in the command history can be configured with the commands:

```
SET PORT HISTORY
SET TTY HISTORY
```

Accessing Telnet Hosts

The asynchronous ports on the router are often used to access Telnet hosts available on the network, using the TELNET command on page 7-11.

To access a Telnet host, use the command:

```
TELNET ipadd
```

where *ipadd* is an IP address in dotted decimal form or a full domain name. For example, a host with the domain name zaphod.beeblebrox.com and IP address 172.16.1.5 can be accessed with either of the commands:

```
TELNET zaphod.beeblebrox.com
TELNET 172.16.1.5
```

If a domain name is specified, the router sends a request to a name server to translate the domain name to an IP address. If the translation is successful, the router attempts to make a connection to the host specified by the IP address.



A name server must be defined with the SET IP NAMESERVER command on page 6-95 of Chapter 6, Internet Protocol (IP). The domain name lookup may take several seconds, during which time the normal router prompt will reappear. When the lookup is complete, a message will be displayed indicating whether or not the lookup was successful.

Popular Telnet hosts can be assigned a short nickname using the command:

```
SET IP HOST=nickname IPADDRESS=ipadd
```

To see the current list of nicknames use the command:

```
SHOW IP HOST
```

The Telnet host can now be accessed with the command:

```
TELNET nickname
```

For example, if the Telnet host zaphod.beeblebrox.com is assigned the nickname zaphod, then it can be accessed with the command:

```
TELNET zaphod
```

Using nicknames solves two problems: it reduces the time delay associated with domain name lookups, and it saves users having to remember IP addresses or long domain names.

See the SET IP HOST command on page 6-91 of *Chapter 6, Internet Protocol (IP)* and the SHOW IP HOST command on page 6-120 of *Chapter 6, Internet Protocol (IP)* for more information about configuring host names.

Command Reference

This section describes the commands available on the router to configure and use the terminal server functions on the router.

When a user executes the TELNET command on page 7-11 to access a service on the network, the port to which the user's terminal is connected is said to be *assigned*. Most of the commands for configuring an asynchronous terminal port described in this section do not work when the port is assigned. An error message to that affect will be displayed if an attempt is made to change the characteristics of a port that is currently assigned.

See "Conventions" on page xxxv of *Preface* in the front of this manual for details of the conventions used to describe command syntax. See *Appendix A, Messages* for a complete list of messages and their meanings.

SET TELNET

Syntax SET TELNET [TERMTYPE=*termstring*] [INSERTNULL={ON|OFF}]

where:

- *termstring* is a character string, 1 to 31 characters in length. If the string contains spaces it must be enclosed in double quotes.

Description This command sets the terminal type string used, and the null insertion behaviour for all outgoing Telnet sessions.

The TERMTYPE parameter specifies a terminal identification string that is passed to a remote Telnet server upon connection. The default option is the string "UNKNOWN". The terminal identification is usually used by the remote system to set the terminal attributes for the Telnet session.

The INSERTNULL parameter, when set to ON, specifies that a NULL character should be inserted after each CR sent to the remote host. The default is OFF.

Examples To set the terminal identification string to vt100, use the command:

```
SET TELNET TERMTYPE=vt100
```

See Also TELNET

SET TTY

Syntax SET TTY [HISTORY=0..99] [PAGE=4..99] [PROMPT={*string-15* | DEFAULT | OFF}] [TYPE={DUMB | VT100}]

where:

- *prompt* is a character string, 1 to 15 characters in length. If the string contains spaces it must be enclosed in double quotes. The string is not case sensitive.

Description This command sets the default values for TTY devices created for Telnet connections. Multiple options may be specified in the same command.



To change the settings for a Telnet connection immediately, use the SET PORT command on page 2-19 of Chapter 2, Interfaces.

The HISTORY parameter sets the number of commands saved in the command history for future recall. The minimum number is 0 and the maximum is 99. Setting the history length to zero for a port does not clear all the commands from the history. The command history is cleared with the RESET PORT HISTORY command on page 2-18 of *Chapter 2, Interfaces*. The default history length for asynchronous ports and Telnet connections is 30.

The PAGE parameter sets the number of lines of command output displayed on the terminal screen before the router pauses and waits for the user to press a key to continue. This number may range from 4 to 99. The default is 22 for both asynchronous ports and Telnet connections. If PAGE is set to OFF, paging is disabled.

The PROMPT parameter sets the prompt for the port to either the default string, such as:

```
CMD>
```

or a user-specified string, or disables the prompt. It is often convenient to disable the prompt if the port is being used as a manager port or for debugging network problems, as it reduces the clutter on the terminal screen. This option only has effect when the port is not assigned. When the port is assigned, prompting is controlled by the host.

The TYPE parameter specifies the type of terminal attached to the port. If TYPE is set to VT100 the router expects the terminal to support standard VT100 escape sequences, and will use them. If TYPE is set to DUMB, the router will not use VT100 escape sequences. The DUMB option is only required for Telnet clients that do not support VT100 escape sequences. The default is VT100 for both asynchronous ports and Telnet connections.

Examples To set PAGE mode off for all subsequent Telnet connections, use the command:

```
SET TTY PAGE=OFF
```

See Also SET PORT in *Chapter 2, Interfaces*
SHOW PORT in *Chapter 2, Interfaces*
SHOW TTY

SHOW TTY

Syntax `SHOW TTY [=tty-number|ALL] [{SUMMARY|DEFAULT}]`

where:

- *tty-number* is the number of a TTY device.

Description This command displays information about one or all of the TTY devices defined on the router at the time the command is issued. There is a TTY device dedicated to each port which is always present. Other TTY devices are created and destroyed as they are required for Telnet logins and multiple sessions.

If a TTY number is specified then only the information for that TTY is displayed. If a TTY number is not specified then information for the TTY where the command is issued is displayed. If ALL is specified then information for all the TTYs on the router is displayed.

If no other parameters are specified then full configuration information for the specified TTY is displayed (Figure 7-2 on page 7-8, Table 7-3 on page 7-9). The SUMMARY parameter generates an abbreviated one-line display for each TTY specified (Figure 7-3 on page 7-9, Table 7-4 on page 7-10). The DEFAULT parameter displays the default values assigned to TTY devices created for Telnet connections (Figure 7-4 on page 7-10, Table 7-5 on page 7-10). A TTY number may not be specified with the DEFAULT parameter.

If the command is issued from a connection with USER privilege the TTY number may not be specified and the information displayed is for the TTY from which the command was issued.

Figure 7-2: Example output from the SHOW TTY command.

```
TTY information
Instance ..... 30
Login name ..... manager
Description ..... Telnet 1
Secure ..... yes
Connections to ..... 21
Current connection ..... 0
In flow state ..... on
Out flow state ..... on
Attached module ..... Telnet
Attached module instance .. 1
Type ..... VT100
Prompt ..... default
Echo ..... yes
Attention ..... char
Manager ..... yes
Edit mode ..... insert
History length ..... 30
Page mode/length ..... 22
```

Table 7-3: Parameters displayed in the output of the SHOW TTY command.

Parameter	Meaning
Instance	The instance number for the TTY device.
Login name	The login name of the user logged in to this port (if any).
Description	The name assigned to the port.
Secure	Whether or not the port is secure; one of "yes" or "no".
Connections to	A list of TTY devices (if any) to which this port TTY is linked for the purpose of providing multiple sessions.
Current connection	The instance number of the TTY that this port TTY is currently connected to, or "none" if there is no active connection.
In flow state	The input flow control state for the TTY device; one of "on" or "off".
Out flow state	The output flow control state for the TTY device; one of "on" or "off".
Attached module	The name of the module that owns the TTY; by default this will be "TSER" (Terminal Server).
Attached module instance	The instance of the module that owns the TTY.
Type	The terminal type setting for the TTY; one of "dumb" or "VT100".
Prompt	The prompt for this TTY; one of "default", "off", "login", "password", "confirm", "encapsulation", or a user-defined string.
Echo	Whether or not the TTY echoes characters received; one of "yes" or "no".
Attention	The attention character for this TTY; one of "none", "break" or "char".
Manager	Whether or not the TTY has MANAGER privilege; one of "yes" or "no".
Edit mode	The edit mode for the TTY; one of "?", "insert" or "overstrike".
History length	The maximum number of commands that will be held in the command history for this TTY.
Page mode/length	The number of lines of command output the router will display before pausing and waiting for the user to press a key, or "off" if page mode is disabled for this TTY.

Figure 7-3: Example output from the SHOW TTY=ALL SUMMARY command.

TTY Description	User name	Module	Inst	Mgr
016 Port 0	support	TSER	000	yes
018 Telnet 1	manager	TELN	001	yes

Table 7-4: Parameters displayed in the output of the SHOW TTY=ALL SUMMARY command.

Parameter	Meaning
TTY	The instance number of the TTY
Description	The name of the port, for a TTY dedicated to a port. For a Telnet login TTY the description is "Telnet" followed by the Telnet instance number.
User name	The login name of the user logged in to the TTY (if any).
Module	The name of the module that is connected to the TTY.
Inst	The instance number of the module that is connected to the TTY.
Mgr	Whether or not the TTY has MANAGER privilege; one of "yes" or "no".

Figure 7-4: Example output from the SHOW TTY DEFAULT command.

```

TTY Default Settings
-----
History length.....20
Page length.....22
Prompt.....default
Type.....VT100

```

Table 7-5: Parameters displayed in the output of the SHOW TTY DEFAULT command.

Parameter	Meaning
History length	The default maximum number of commands that will be held in the command history for a TTY.
Page mode/length	The default number of lines of command output the router will display before pausing and waiting for the user to press a key, or "off" if page mode is disabled for a TTY.
Prompt	The default prompt for a TTY; one of "default", "off", "login", "password", "confirm", "encapsulation", or a user-defined string.
Type	The default terminal type setting for a TTY; one of "dumb" or "VT100".

Examples To display the TTY configuration for a Telnet connection, use the command:

```
SHOW TTY
```

To display a summary of all the TTY information for a router, use the command:

```
SHOW TTY=ALL SUMMARY
```

See Also SET PORT in *Chapter 2, Interfaces*
 SET TTY
 SHOW PORT in *Chapter 2, Interfaces*

TELNET

Syntax TELNET {*ipadd*|*host*}

where:

- *ipadd* is an IP address in dotted decimal notation.
- *host* is a full domain name of a host, a host nickname created with the SET IP HOST command on page 6-91 of *Chapter 6, Internet Protocol (IP)*, or a host name in the same domain.

Description This command attempts to open a Telnet connection to a Telnet host at the specified IP address or with the specified name. If the command is successful then the message in Figure 7-5 on page 7-11 will be followed by the host prompt. When the user logs off from the host the connection is terminated and the router prompt reappears. The Telnet session can also be terminated by pressing [Ctrl/D].

If the *sysName* MIB object is set to the router's fully qualified domain name (e.g. *router.company.com*) using the SET SYSTEM NAME command on page 1-55 of *Chapter 1, Operation*, and a name server has been defined using the SET IP NAMESERVER command on page 6-95 of *Chapter 6, Internet Protocol (IP)*, then the command:

```
TELNET mainhost
```

will attempt a Telnet connection to the host "mainhost.company.com", provided "mainhost" is not an IP nickname (IP nicknames take precedence).

Figure 7-5: Example output from the TELNET command.

```
TELNET. Attempting to connect to 192.168.35.17, please wait...
```



If a domain name is specified, the router sends a request to a name server to translate the domain name into an IP address. This may take several seconds during which time the normal router prompt will reappear. When the name server responds (or fails to respond), a message will be displayed indicating that either the lookup was unsuccessful, or that it was successful and an attempt is being made to connect to a host at the specified IP address.



A user is permitted to issue the TELNET command only if the user has the TELNET attribute set to YES in the user database. See Chapter 1, Operation for further information on these security features.



If a user Telnets to the router but does not attempt to login within one minute, the router automatically times out the session and terminates the Telnet connection.

Examples To connect to Telnet host *zaphod.beeblebrox.com* use the command:

```
TELNET zaphod.beeblebrox.com
```

See Also ADD IP HOST in *Chapter 6, Internet Protocol (IP)*
 DELETE IP HOST in *Chapter 6, Internet Protocol (IP)*
 SET IP HOST in *Chapter 6, Internet Protocol (IP)*
 SET IP NAMESERVER in *Chapter 6, Internet Protocol (IP)*
 SET SYSTEM NAME in *Chapter 1, Operation*
 SET TELNET
 SHOW IP HOST (in *Chapter 6, Internet Protocol (IP)*)

Chapter 8

Compression Services

Introduction	8-2
Data Compression	8-2
ENCO Services	8-4
Compression	8-4
User Modules	8-5
PPP	8-5
X.25 Link Compression	8-5
Command Reference	8-5
DISABLE ENCO COMPSTATISTICS	8-5
DISABLE ENCO DEBUGGING	8-6
ENABLE ENCO COMPSTATISTICS	8-6
ENABLE ENCO DEBUGGING	8-7
RESET ENCO COUNTERS	8-7
SET ENCO SW	8-7
SHOW ENCO	8-8
SHOW ENCO CHANNEL	8-9
SHOW ENCO COUNTERS	8-13

Introduction

This chapter describes the data compression services available on the router, how the services are provided, the router network functions which use these services, and how to monitor the services.

The ENCO module provides data compression services to other router software modules (referred to as user modules). See “*Data Compression*” on page 8-2 for an overview of these services. See “*ENCO Services*” on page 8-4 for a description of how these services are provided by the ENCO module.

Data Compression

Data compression for routers is driven by the high cost of *wide area network* (WAN) access and user demands for increased bandwidth. The cost of WAN access is a significant part of the cost of providing a data network and the use of data compression on networks can result in significant savings.

Compression increases the effective throughput of data across a network link by reducing the size of packets. This allows more packets to be transmitted over the link in the same time interval, or the same number of packets to be transmitted over a slower (and cheaper) link in the same time interval.

Data compression works by identifying redundancy in the data and producing an encoded form which is smaller yet contains all the information required to recreate the original data. This is called *lossless data compression*, as opposed to voice and video compression which, due to their analog nature, normally use *lossy data compression* algorithms. Most modern high performance data compression techniques use variations of the Lempel-Ziv algorithm. This algorithm compresses data by maintaining data histories at the compression and decompression ends of the link. These histories contain the most recent data which has been transmitted on a data link. The data to be compressed is compared against the history to find any common sequences. If a match is found, a reference to the position of the matching sequence in the history is sent instead of the data sequence itself. Compression is achieved because the reference is smaller than the sequence it represents. The algorithm is adaptive, adjusting automatically to produce the best compression ratio for the content of the data being compressed. Typically a checksum is also added to the data before compression to allow the validity of the data to be checked on decompression.



It is impossible to compress all possible data streams. A stream of totally random data has no redundancy and therefore can not be compressed. Similarly, data which has already been compressed is unlikely to be further compressed. Some compression algorithms, such as the STAC LZS algorithm cause uncompressable data to expand during the compression process.

Network data compression can be divided into three categories—link compression, payload compression and header compression.

Link Compression

Link compression has traditionally been provided by an external device connected between a router port and the WAN access device. The main disadvantages of external compression devices are that they require a separate

connection to each router port requiring compression and to each WAN access device, and they can not be managed from within the router's management structure.

Link compression operates by compressing the whole data stream, including the network layer packet headers used for routing. This means that the packet header is no longer accessible by intermediate routers which do not support the particular compression algorithm. Even if an intermediate router does support the particular compression algorithm, packets must be decompressed and re-compressed at each router so that the packet headers can be read. This places an additional load on the router and results in high latency. Consequently, external link compression is normally only used in point-to-point configurations where the local and remote routers are directly connected, without any intermediate routers.

Integrating the compression function into the router enables a single compression resource to support the compression of multiple links over any router interface, replacing multiple external compression devices. Integration also allows the router to support protocols such as PPP multilink, which can spread data from one compression channel across multiple physical links. The compression process can be configured and monitored using the router's own management interface, instead of a separate management system used only for the external compression device.

See *Chapter 18, Link Compression* for more information about configuring link compression on the router.

Payload Compression

Payload compression is used to compress packet data at the network layer, without changing the packet header. Since the routing information remains unchanged the packet can be carried across a routed network, such as the Internet, without requiring the intermediate routers to support the compression algorithm or have any knowledge about how to access the compressed data.

Payload compression is usually not as efficient as link compression due to the fact that each packet must be compressed with no reference to any other packet—as packets may be lost or re-ordered while traversing the network. This means that the compression history must be cleared before compressing a packet, losing any advantage gained from compressing the previous packets. Only large packets or packets containing highly compressible data benefit greatly from payload compression.

Header Compression

Van Jacobson's header compression algorithm (defined in RFC 1144) can be used in TCP/IP networks to compress the standard 40-byte TCP/IP header of TCP packets down to 5 bytes. This produces a significant performance improvement when the majority of traffic consists of small packets. Because the router processor must perform the compression calculations this method is normally recommended for lower speed (less than 64 kbps) links.

Van Jacobson's header compression applies only to TCP packets carried over Point-to-Point Protocol (PPP) links. The ENCO module is not required for header compression.

See *Chapter 6, Internet Protocol (IP)* for more information about configuring Van Jacobson's header compression.

ENCO Services

The ENCO module provides compression services to user modules via channel pairs. A user module requests a service, specifying any configuration needed for the service, and is attached to an ENCO channel pair if the service and free channels are available. A channel pair consists of an encoding channel and a decoding channel. An encoding channel is used for compression. A decoding channel is used for decompression.

The command:

```
SHOW ENCO
```

displays general information about the ENCO module and the services that are available.

A user module which requests the retention of process histories between packets for a compression service (see *"User Modules"* on page 8-5) may also request that the history of one of its channels be reset. Whenever a decoding channel gets out of step with its associated encoding channel the encoding channel's history must be reset.

Compression

STAC LZS compression is provided in software. To use software compression, the number of channels required must be configured using the command:

```
SET ENCO SW STACCHANNELS=0..4
```

This command must be run from the boot configuration script since the memory required by the software compression algorithms must be contiguous and the most efficient way to acquire contiguous memory is just after the router has rebooted. The maximum number of software compression channels is limited due to the large amount of memory required by the software compression algorithm; STAC LZS requires 13 KBytes per channel. The limit is four STAC LZS channels. By default no compression channels are configured.

For STAC LZS compression, it is possible to select between compression speed and compression ratio using the command:

```
SET ENCO SW STACSPEED=0..3
```

Setting STACSPEED to a high value minimises compression time at the expense of the compression ratio and is most suitable for high aggregate line speeds. Setting STACSPEED to a low value maximises the compression ratio at the expense of processing time and is most effective for low aggregate link speeds. Intermediate values give a balance between the two. The STACSPEED value only effects the transmit (compression) path; the receive (decompression) path is not affected by the value of STACSPEED.

User Modules

PPP

The PPP module can use the services of the ENCO module to provide link compression.

The router implements the Compression Control Protocol (CCP) as defined by RFC 1962 to provide compression on PPP. CCP provides a method for negotiating the compression algorithm to use and algorithm-specific parameters such as the check mode. It also provides a mechanism for synchronising the compression histories at each end of the link if they become unsynchronised. The use of STAC LZS compression with PPP is defined in RFC 1962.

For more information about configuring PPP link compression see *Chapter 18, Link Compression*.

X.25 Link Compression

The router uses a simple static configuration process to provide STAC LZS compression for X.25. X.25 does not reset compression links as it is a reliable transmission protocol. For more information about configuring X.25 compression see *Chapter 18, Link Compression*.

Command Reference

This section describes the commands available on the router to configure and monitor the compression processes on the router.

For each interface over which compression is to be used, a higher layer module must be configured to use compression. Compression is supported on Point-to-Point Protocol (PPP) and X.25 interfaces. See *Chapter 3, Point-to-Point Protocol (PPP)* for details of the commands required to enable compression on a PPP interface. See *Chapter 5, X.25* for details of the commands required to enable compression on an X.25 interface. See *Chapter 18, Link Compression* for more information about configuring link compression..

See “Conventions” on page xxxv of *Preface* in the front of this manual for details of the conventions used to describe command syntax. See *Appendix A, Messages* for a complete list of messages and their meanings.

DISABLE ENCO COMPSTATISTICS

Syntax `DISABLE ENCO COMPSTATISTICS`

Description This command disables the calculation and storage of compression ratio statistics for any compression-only ENCO channels.

See Also ENABLE ENCO COMPSTATISTICS
 SHOW ENCO
 SHOW ENCO CHANNEL

DISABLE ENCO DEBUGGING

Syntax DISABLE ENCO DEBUGGING=PACKET

Description This command disables the specified debugging option for the ENCO module. The specified debugging option must currently be enabled. Debugging information is sent to the terminal from which the command was entered. Any combination of options can be disabled using successive commands.

The DEBUG parameter specifies which debugging option is to be disabled. PACKET debugging displays the contents of packets processed by the ENCO module.

Examples To disable the debugging of the contents of packets processed by the ENCO module, use the command:

```
DISABLE ENCO DEBUG=PACKET
```

See Also ENABLE ENCO DEBUGGING

ENABLE ENCO COMPSTATISTICS

Syntax ENABLE ENCO COMPSTATISTICS

Description This command enables the calculation and storage of compression ratio statistics for any compression-only ENCO channels. The collected statistics are displayed in the output of the SHOW ENCO CHANNEL command on page 8-9.

See Also DISABLE ENCO COMPSTATISTICS
 SHOW ENCO

ENABLE ENCO DEBUGGING

Syntax `ENABLE ENCO DEBUGGING=PACKET`

Description This command enables the specified debugging option for the ENCO module. The specified debugging option must currently be disabled. Debugging information is sent to the terminal from which the command was entered. Any combination of options can be enabled using successive commands.

The DEBUG parameter specifies which debugging option is to be enabled. PACKET debugging displays the contents of packets processed by the ENCO module.

Examples To enable the debugging of the contents of packets processed by the ENCO module, use the command:

```
ENABLE ENCO DEBUG=PACKET
```

See Also `DISABLE ENCO DEBUGGING`

RESET ENCO COUNTERS

Syntax `RESET ENCO COUNTERS={JOBPROCESSING|STAC|USER|UTIL}`

Description This command clears general and process-specific counters for the ENCO module. The COUNTER parameter specifies the category of counters to be cleared. If a category is not specified, all ENCO counters are cleared. USER, UTIL and JOBPROCESSING counters display information about the general operation of the ENCO module. The STAC counters display information for the STAC process.

If JOBPROCESSING is specified, counters for the jobs that have been or are still being processed by the ENCO module are reset. If STAC is specified, counters for the STAC compression process are reset. If USER is specified, counters for the interface between the ENCO module and other user modules who use ENCO channels are reset. If UTIL is specified, counters for the interface between the ENCO module and other user modules who use the ENCO module for one-off jobs are reset.

Examples To reset all the ENCO counters, use the command:

```
RESET ENCO COUNTERS
```

See Also `SHOW ENCO COUNTERS`

SET ENCO SW

Syntax `SET ENCO SW [STACCHANNELS=0..4] [STAC SPEED=0..3]`

Description This command changes the configuration parameters for software compression.

The STACCHANNELS parameter specifies the number of STAC LZS compression channels to allocate. Each STAC LZS compression channel requires 13 KBytes of contiguous memory.

The STACSPEED parameter specifies a performance level between 0 (maximum compression ratio) and 3 (maximum compression speed). A high value minimises compression time at the expense of the compression ratio, and is suitable for high aggregate line speeds. A low value maximises the compression ratio and the expense of processing time, and is most effective for low aggregate line speeds. Intermediate values give a balance between the two. The speed setting only effects the transmit (compression) path; the receive (decompression) path does not require a speed parameter.

Compression speed should be set according to the bandwidth of the links over which software compression will be used. Suggested speed settings for different line rates are shown in Table 8-1 on page 8-8.

Table 8-1: Suggested software compression speed settings for different transmission line speeds.

Aggregate Line Speed (kbps)	Compression Speed Setting
> 85	3
60–85	2
40–59	1
< 40	0

Examples To configure three STAC LZS software compression channels, use the command:

```
SET ENCO SW STACCHANNELS=3
```

See Also SHOW ENCO

SHOW ENCO

Syntax SHOW ENCO

Description This command displays information about the ENCO module (Figure 8-1 on page 8-8, Table 8-2 on page 8-9).

Figure 8-1: Example output from the SHOW ENCO command.

```
ENCO Module Configuration

Lowest valid channel ..... 1
Highest valid channel ..... 127
Compression Statistics Enabled ..... FALSE

SW Processes available
  DMAN - Data Manipulation
  STAC - Stac Compression
```

Table 8-2: Parameters displayed in the output of the SHOW ENCO command.

Parameter	Meaning
Lowest valid channel	The identification number of the lowest channel available for use by a user module.
Highest valid channel	The identification number of the highest channel available for use by a user module.
Compression Statistics Enabled	Whether or not gathering of compression statistics is enabled for all channels; one of "TRUE" or "FALSE".
SW Processes available	A list of the software-based processes available to the ENCO module for processing user data packets; one or more of "NONE", "DMAN", or "STAC".

See Also SHOW ENCO CHANNEL
SHOW ENCO COUNTERS

SHOW ENCO CHANNEL

Syntax SHOW ENCO CHANNEL [=channel [COUNTERS]]

where:

- *channel* is a number in the range 0 to 127.

Description This command displays information about active ENCO module channels. If an ENCO channel is not specified, a summary of all currently active channels is displayed (Figure 8-2 on page 8-9, Table 8-3 on page 8-9). If an ENCO channel is specified, detailed configuration and status information about the specified channel is displayed (Figure 8-3 on page 8-10, Table 8-4 on page 8-10). If compression statistics are enabled, the display will include compression statistics.

If the COUNTERS parameter is specified, information counters for the specified channel are displayed (Figure 8-4 on page 8-11, Table 8-5 on page 8-12).

Figure 8-2: Example output from the SHOW ENCO CHANNEL command.

Channel	State	User	UserID	MDL	pktOverhead	Process
2	UP	PPP	00000001	1500	64	STAC

Table 8-3: Parameters displayed in the output of the SHOW ENCO CHANNEL command.

Parameter	Meaning
Channel	The channel identification number.
State	The state of the channel; one of "UP" or "DOWN".

Table 8-3: Parameters displayed in the output of the SHOW ENCO CHANNEL command. (Continued)

Parameter	Meaning
User	The user module attached to this channel; one of "PPP", "MIOX", or "TEST".
UserID	A number used by the user module to identify this channel.
MDL	The maximum data length of packets accepted on this channel.
pktOverhead	The number of bytes that the user module requested be reserved in a packet in front of encoded data.
Process	The process for which the channel is configured; "STAC".

Figure 8-3: Example output from the SHOW ENCO CHANNEL command for a specified channel.

```

Channel ..... 1

Type ..... ENCODE/DECODE
State ..... UP
User ..... PPP
User ID ..... 00000001
Maximum Data Length ..... 1584
Packet Overhead ..... 16
Process ..... STAC
Process Configuration:
  Check Type.....
  Channel Type.....ENCODE/DECODE

```

Table 8-4: Parameters displayed in the output of the SHOW ENCO CHANNEL command for a specified channel.

Parameter	Meaning
Channel	The identification number of the channel.
Type	The mode of the channel; one of "ENCODE/DECODE", "ENCODE ONLY" or "DECODE ONLY".
State	The state of the channel; one of "UP" or "DOWN".
User	The user module attached to this channel; one of "PPP" or "MIOX".
User ID	A number used by the user module to identify this channel.
Maximum Data Length	The maximum data length of packets accepted on this channel.
Packet Overhead	The number of bytes reserved at the head of data packets in front of the encoded data, for lower layer packet headers.
Process	The process for which the channel is configured; "STAC".
Process Configuration	Details about a particular process. The fields displayed vary depending on the process.
Max Data Length	The maximum allowed length of data packets on the channel.

Table 8-4: Parameters displayed in the output of the SHOW ENCO CHANNEL command for a specified channel. (Continued)

Parameter	Meaning
Check Type	The type of checksum to be used; one of "XOR8" or "NONE" (STAC compression).
Compression Statistics	Statistics for the compression process. This section is only displayed when compression statistics have been enabled with the ENABLE ENCO COMPSTATISTICS command on page 8-6.
Number of Packets Compressed	The number of data packets that have been compressed.
Best Compression Ratio	The highest compression ratio achieved.
Mean Compression Ratio	The mean compression ratio achieved.
Worst Compression Ratio	The lowest compression ratio achieved.
Compression Ratio	A range of compression ratios.
Number of Packets	The number of packets compressed, for which the resulting compression ration was in the specified range.

Figure 8-4: Example output from the SHOW ENCO CHANNEL COUNTERS command.

Channel Counters:

UP events	1	DOWN events	0
start config	1	attach good	1
encode NULL packets	0	decode NULL packets	0
encode bad priorities	0	decode bad priorities	0
encode bad length	0	decode bad length	0
encode actions sent	0	decode actions sent	0
good encodes	0	good decodes	0
bad encodes	0	bad decodes	0
reset E actions sent	0	reset D actions sent	0
good encode resets	0	good decode resets	0
bad encode resets	0	bad decode resets	0
discarded encode jobs	0	discarded decode jobs	0

Table 8-5: Parameters displayed in the output of the SHOW CHANNELS COUNTERS command.

Parameter	Meaning
UP events	The number of times the channel has entered the "UP" state.
DOWN events	The number of times the channel has entered the "DOWN" state.
start config	The number of times a configure operation has started on the channel.
attach good	The number of successful attach operations on the channel.
encode NULL packets	The number of encode requests received from a user module with no data packet.
decode NULL packets	The number of decode requests received from a user module with no data packet.
encode bad priorities	The number of encode requests received from a user module with a data packet containing an unknown priority.
decode bad priorities	The number of decode requests received from a user module with a data packet containing an unknown priority.
encode bad length	The number of encode requests received from a user module with a data packet with a bad length.
decode bad length	The number of decode requests received from a user module with a data packet with a bad length.
encode actions sent	The number of encode actions which have been sent to the process on this channel.
decode actions sent	The number of decode actions which have been sent to the process on this channel.
good encodes	The number of successful encode operations on the channel.
good decodes	The number of successful decode operations on the channel.
bad encodes	The number of unsuccessful encode operations on the channel.
bad decodes	The number of unsuccessful decode operations on the channel.
reset E actions sent	The number of encode reset actions which have been sent to the process on the channel.
reset D actions sent	The number of decode reset actions which have been sent to the process on the channel.
good encode resets	The number of successful encode resets on the channel.
good decode resets	The number of successful decode resets on the channel.
bad encode resets	The number of unsuccessful encode resets on the channel.
bad decode resets	The number of unsuccessful decode resets on the channel.
discarded encode jobs	The number of encode jobs discarded due to queue overloading or a channel reset.
discarded decode jobs	The number of decode jobs discarded due to queue overloading or a channel reset.

Examples To show a summary of all active ENCO channels, use the command:

```
SHOW ENCO CHANNEL
```

To show detailed configuration and status information for channel 1, use the command:

```
SHOW ENCO CHANNEL=1
```

To show information counters for channel 1, use the command:

```
SHOW ENCO CHANNEL=1 COUNTERS
```

See Also SHOW ENCO
SHOW ENCO COUNTERS

SHOW ENCO COUNTERS

Syntax SHOW ENCO COUNTERS={JOBPROCESSING|STAC|USER|UTIL}

Description This command displays information counters for the ENCO module. The COUNTER parameter specifies the category of counters to display. The USER, UTIL and JOBPROCESSING counters display information about the general operation of the ENCO module. The STAC counters display information for the STAC process.

If JOBPROCESSING is specified, counters for the jobs that have been or are still being processed by the ENCO module are displayed (Figure 8-5 on page 8-14, Table 8-6 on page 8-15).

If STAC is specified, counters for the STAC compression process are displayed (Figure 8-6 on page 8-16, Table 8-7 on page 8-16).

If USER is specified, counters for the interface between the ENCO module and other user modules who use ENCO channels are displayed (Figure 8-7 on page 8-19, Table 8-8 on page 8-19).

If UTIL is specified, counters for the interface between the ENCO module and other user modules who use the ENCO module for one-off jobs are displayed (Figure 8-8 on page 8-20, Table 8-9 on page 8-21).

Figure 8-5: Example output from the SHOW ENCO COUNTERS=JOBPROCESSING command.

Input queue lengths	
Immediate queue.....	0
Priority queue 0 (high).....	0
Priority queue 1.....	0
Priority queue 2.....	0
Priority queue 3.....	0
Priority queue 4.....	0
Priority queue 5.....	0
Priority queue 6.....	0
Priority queue 7.....	0
Priority queue 8 (low).....	0
Total actions queued.....	0
Lowest Input Priority Queue.....	4
Highest Input Priority Queue.....	0
Input Queue Length Limit.....	30
Input Queue discards	
Immediate queue.....	0
Priority queue 0 (high).....	0
Priority queue 1.....	0
Priority queue 2.....	0
Priority queue 3.....	0
Priority queue 4.....	0
Priority queue 5.....	0
Priority queue 6.....	0
Priority queue 7.....	0
Priority queue 8 (low).....	0
Total Input Queue discards.....	0
Input Queue jobs processed	
Immediate queue.....	4
Priority queue 0 (high).....	0
Priority queue 1.....	0
Priority queue 2.....	0
Priority queue 3.....	0
Priority queue 4.....	310
Priority queue 5.....	0
Priority queue 6.....	0
Priority queue 7.....	0
Priority queue 8 (low).....	0
Total Input Queue jobs processed	314
Output queue length.....	0
Output queue jobs completed.....	310
Output queue discards.....	0

Table 8-6: Parameters displayed in the output of the SHOW ENCO COUNTERS=JOBPROCESSING command.

Parameter	Meaning
Input queue lengths	The lengths of the ENCO module input queues.
Immediate queue	The number of actions on the immediate input queue.
Priority queue n	The number of actions on the specified priority input queue.
Total actions queued	The total number of actions on the input queues.
Lowest Input Priority Queue	The lowest input priority queue with queued actions.
Highest Input Priority Queue	The highest input priority queue with queued actions.
Input Queue Length Limit	The maximum length of the input queues.
Input Queue discards	The numbers of actions discarded from the input queues.
Immediate queue	The number of actions discarded from the immediate input queue.
Priority queue n	The number of actions discarded from the specified priority input queue.
Total Input Queue discards	The total number of actions discarded from the input queues.
Input Queue jobs processes	The numbers of jobs processed from the input queues.
Immediate queue	The number of jobs processed from the immediate input queue.
Priority queue n	The number of jobs processed from the specified priority input queue.
Total Input Queue jobs processed	The total number of jobs processed from the input queues.
Output queue length	The length of the output queue.
Output queue jobs completed	The number of jobs completed off the output queue.
Output queue discards	The number of jobs discarded from the output queue.

Figure 8-6: Example output from the SHOW ENCO COUNTERS=STAC command.

General Enco STAC Counters			
procHandParmBadJobType	0	procHandParmNullCfg	0
procHandParmNullInPkt	0	procHandParmNullOutPkt	0
procHandParmBadMdl	0		
procHandHwCompFail	0	procHandSwCompFail	0
procHandCompNullCheckPt	0	procHandCompGood	0
procHandHwDecompFail	0	procHandSwDecompFail	0
procHandDecompCheckFail	0	procHandDecompGood	0
procHandConfEDone	0	procHandConfDDone	0
procHandResetEDone	0	procHandResetDDone	0
procHandFailReconfJob	0	procHandFailUnknownJob	0
configNoResource	0	configHwNoHistory	0
configSwNoHistory	0	configGood	0
destroyParmNullConfig	0	destroyGood	0
Enco STAC SW Counters			
compParmNullHistoryPt	0	compParmNullSourcePt	0
compParmNullResultPt	0	compParmBadMdl	0
compMdlAbort	0	compError	0
compGood	0		
decompParmNullHistoryPt	0	decompParmNullSourcePt	0
decompParmNullResultPt	0	decompParmBadMdl	0
decompMdlAbort	0	decompDestExhaust	0
decompEocMissing	0	decompError	0
decompGood	0		
resetParmNullHistoryPt	0		
resetDecompParmNullHistPt	0	resetDecompGood	0
resetDecompError	0		
dummyCompParmNullHistoryPt	0	dummyCompError	0
dummyCompGood	0		
setSpeedParmBadSpeed	0		
allocHistParmBadChan	0	allocHistChainNotContig	0
allocHistChainTooShort	0	allocHistGood	0

Table 8-7: Parameters displayed in the output of the SHOW ENCO COUNTERS=STAC command.

Parameter	Meaning
procHandParmBadJobType	The number of times the STAC process handler received a bad job type parameter.
procHandParmNullCfg	The number of times the STAC process handler received a job with a NULL STAC configuration.
procHandParmNullInPkt	The number of times the STAC process handler received a job with a NULL In Packet.
procHandParmNullOutPkt	The number of times the STAC process handler received a job with a NULL Out Packet.
procHandParmBadMdl	The number of times the STAC process handler received a job with an invalid Maximum Data Length value.
procHandHwCompFail	The number of times the STAC process handler had a failed hardware compression job.
procHandSwCompFail	The number of times the STAC process handler had a failed software compression job.
procHandCompNullCheckPt	The number of times the STAC process handler received a job with a NULL check pointer.

Table 8-7: Parameters displayed in the output of the SHOW ENCO COUNTERS=STAC command. (Continued)

Parameter	Meaning
procHandCompGood	The number of times the STAC process handler had a successful compression job.
procHandHwDecompFail	The number of times the STAC process handler had a failed hardware decompression job.
procHandSwDecompFail	The number of times the STAC process handler had a failed software decompression job.
procHandDecompCheckFail	The number of times the STAC process handler had a software decompression job which had a bad checksum.
procHandDecompGood	The number of times the STAC process handler had a successful decompression job.
procHandConfEDone	The number of times the STAC process handler had a successful compression channel configure job.
procHandConfDDone	The number of times the STAC process handler had a successful decompression channel configure job.
procHandResetEDone	The number of times the STAC process handler had a successful compression channel reset job.
procHandResetDDone	The number of times the STAC process handler had a successful decompression channel reset job.
procHandFailReconfJob	The number of times the STAC process handler had a failed channel reconfigure job.
procHandFailUnknownJob	The number of times the STAC process handler failed because it received an unknown job.
configNoResource	The number of times a STAC channel configure failed because there were no ENCO channels available.
configHwNoHistory	The number of times a STAC channel configure failed because there were no hardware histories available.
configSwNoHistory	The number of times a STAC channel configure failed because there were no software histories available.
configGood	The number of successful STAC channel configure jobs.
destroyParmNullConfig	The number of times a STAC channel destroy failed because the channel did not exist.
destroyGood	The number of successful STAC channel destroy jobs.
compParmNullHistoryPt	The number of times a software compression job was received with a NULL history pointer.
compParmNullSourcePt	The number of times a software compression job was received with a NULL source packet pointer.
compParmNullResultPt	The number of times a software compression job was received with a NULL result packet pointer.
compParmBadMdl	The number of times a software compression job was received with an invalid maximum data length value.
compMdlAbort	The number of times a software compression job failed due to the maximum data length being exceeded.
compError	The number of times a software compression job failed due to an unspecified error.
compGood	The number of successful software compression jobs.

Table 8-7: Parameters displayed in the output of the SHOW ENCO COUNTERS=STAC command. (Continued)

Parameter	Meaning
decompParmNullHistoryPt	The number of times a software decompression job was received with a NULL history pointer.
decompParmNullSourcePt	The number of times a software decompression job was received with a NULL source packet pointer.
decompParmNullResultPt	The number of times a software decompression job was received with a NULL result packet pointer.
decompParmBadMdl	The number of times a software decompression job was received with an invalid maximum data length value.
decompMdlAbort	The number of times a software decompression job failed due to the maximum data length being exceeded.
decompDestExhaust	The number of times a software decompression job failed due to the result data being exhausted.
decompEocMissing	The number of times a software decompression job failed due to the failure to find an end of compressed data marker.
decompError	The number of times a software decompression job failed due to an unspecified error.
decompGood	The number of successful software decompression jobs.
resetParmNullHistoryPt	The number of times a software compression channel reset job was received with a NULL history pointer.
resetDecompParmNullHistoryPt	The number of times a software decompression channel reset job was received with a NULL history pointer.
resetDecompGood	The number of successful software decompression channel resets.
resetDecompError	The number of failed software decompression channel resets.
dummyCompParmNullHistoryPt	The number of times a software compression channel dummy compression job was received with a NULL history pointer.
dummyCompError	The number of failed software compression channel dummy compressions.
dummyCompGood	The number of successful software compression channel dummy compressions.
setSpeedParmBadSpeed	The number of times a set software compression speed job was received with an invalid speed value.
allocHistParmBadChan	The number of times an allocate software compression history job was received for an invalid channel.
allocHistChainNotContig	The number of times an allocate software compression history job failed due to a non-contiguous memory allocation.
allocHistChainTooShort	The number of times an allocate software compression history job failed because it did not have enough non-contiguous memory.
allocHistGood	The number of successful allocate software compression history jobs.

Figure 8-7: Example output from the SHOW ENCO COUNTERS=USER command.

ENCO User Interface Counters:			
startConfig	2	startReconfig	0
attachGood	2	attachFail	0
attachNoConfig	0	attachBadUserType	0
attachedInvalidChannel	0	attachedUnusedChannel	0
attachProcNotAvail	0		
reconfigInvalidChannel	0	reconfigUnusedChannel	0
reconfigNoConfig	0		
detachInvalidChannel	0	detachUnusedChannel	0
detachedInvalidChannel	0	detachedUnusedChannel	0
detachGood	0		
decodeInvalidChannel	0	decodeUnusedChannel	0
encodeInvalidChannel	0	encodeUnusedChannel	0
codedInvalidChannel	0	codedUnusedChannel	0
resetInvalidChannel	0	resetUnusedChannel	0
resetDoneInvalidChannel	0	resetDoneUnusedChannel	0
configBadMode	0	configBadUserType	0
configBadPktLength	0		
configBadCompType	0	configBadHistoryMode	0
configBadCheckType	0		
discardInvalidChannel	0	discardUnusedChannel	0

Table 8-8: Parameters displayed in the output of the SHOW ENCO COUNTERS=USER command.

Parameter	Meaning
startConfig	The number of channel configuration requests initiated.
startReconfig	The number of channel reconfiguration requests started.
attachGood	The number of successful channel attaches.
attachFail	The number of unsuccessful channel attaches.
attachNoConfig	The number of channel attach requests received with no configuration information.
attachBadUserType	The number of channel attach requests containing a bad user type.
attachedInvalidChannels	The number of channel attached events on invalid channels.
attachedUnusedChannel	The number of channel attached events on unused channels.
attachProcNotAvail	The number of attaches requesting a process while the process is unavailable.
reconfigInvalidChannel	The number of channel reconfigure requests on invalid channels.
reconfigUnusedChannel	The number of channel reconfigure requests on unused channels.
reconfigNoConfig	The number of channel reconfigure requests received with no configuration information.
detachInvalidChannel	The number of channel detach requests on nonexistent channels.

Table 8-8: Parameters displayed in the output of the SHOW ENCO COUNTERS=USER command. (Continued)

Parameter	Meaning
detachedInvalidChannel	The number of channel detached events on invalid channels.
detachedUnusedChannel	The number of channel detached events on unused channels.
detachGood	The number of successful channel detaches.
decodeInvalidChannel	The number of decode requests on nonexistent channels.
decodeUnusedChannel	The number of decode requests on unused channels.
encodeInvalidChannel	The number of encode requests on nonexistent channels.
encodeUnusedChannel	The number of encode requests on unused channels.
codedInvalidChannel	The number of encode events on nonexistent channels.
codedUnusedChannel	The number of encode events on unused channels.
resetInvalidChannel	The number of channel reset requests on nonexistent channels.
resetUnusedChannel	The number of channel reset requests on unused channels.
resetDoneInvalidChannel	The number of channel reset requests on invalid channels.
resetDoneUnusedChannel	The number of channel reset requests on nonexistent channels.
configBadMode	The number of channel configuration requests containing a bad mode.
configBadUserType	The number of channel configuration requests containing a bad user type.
configBadPktLength	The number of channel configuration requests containing a bad packet length.
configBadCompType	The number of channel configuration requests containing a bad compression type.
configBadHistoryMode	The number of channel configuration requests containing a bad history mode.
configBadCheckType	The number of channel configuration requests containing a check type.
discardInvalidChannel	The number of discarded jobs on invalid channels.
discardUnusedChannel	The number of discarded jobs on nonexistent channels.

Figure 8-8: Example output from the SHOW ENCO COUNTERS=UTIL command.

ENCO Utility Counters:

codeNullPacket	0	codeBadPacketPriority	0
codeBadPacketLength	0	codeBadConfig	0
actionSentEncode	0	actionSentDecode	0
configureGood	2	configureFail	0
encodeGood	0	decodeGood	2
encodeBad	0	decodeBad	0

Table 8-9: Parameters displayed in the output of the SHOW ENCO COUNTERS=UTIL command.

Parameter	Meaning
codeNullPacket	The number of utility jobs where the packet to be processed was null.
codeBadPacketLength	The number of utility jobs where the packet to be processed had a bad packet length.
actionSentEncode	The number of encode jobs started.
configureGood	The number of successful attempts to configure the utility channel.
encodeGood	The number of completed encode jobs.
encodeBad	The number of failed encode jobs.
codeBadPacketPriority	The number of utility jobs where the packet to be processed had a bad priority.
codeBadConfig	The number of utility jobs where the configuration was invalid.
actionSentDecode	The number of decode jobs started.
configureFail	The number of failed attempts to configure the utility channel.
decodeGood	The number of completed decode jobs.
decodeBad	The number of failed decode jobs.

Chapter 9

Test Facility

Introduction	9-2
Ethernet Port Tests	9-4
Asynchronous Port Tests	9-6
Basic Rate ISDN Port Tests	9-7
Command Reference	9-8
DISABLE TEST INTERFACE	9-8
ENABLE TEST INTERFACE	9-9
RESET TEST INTERFACE	9-10
SHOW TEST	9-10

Introduction

This chapter describes the main features of the Test Facility on the router, and how to set up and use the Test Facility. The Test Facility is intended to be used by the manufacturer during router production and servicing, and by distributors and router owners to verify router operation.

The Test Facility provides a simple, efficient method of validating the operation of the router hardware, including the router interfaces (Ethernet, BRI, etc.). The router processing core is not tested by the Test Facility since it must be operational for the Test Facility to operate. The processing core is tested during every power up.

The Test Facility runs in the normal router operating system environment. This means that the router processing core and an access port must be operational before testing begins. The tests operate by using standard router device drivers, so this software must also be fully operational. Tests are controlled using the router command interface; either from a local terminal port or remotely using Reverse Telnet. An SNMP management system can determine whether a port is being tested, but cannot be used to initiate a test; the section of the MIB used to set a port to test mode may be written and read, but will not result in any action being taken.

Tests on interfaces require external connections to be made to loopbacks or specialised test hardware.

When a test is initiated from a local asynchronous connection, test messages will be printed for tests which are completed or halted. These messages may occur at any time during the test. If the test command has been entered from another source, such as a remote Reverse Telnet connection, these messages are not printed. In this case the test status is only printed in response to a SHOW TEST command on page 9-10.



Tests have the potential to generate network problems if tests are enabled on active resources (for example, interfaces connected to a LAN). To limit this potential problem, tests automatically halt if an active resource is detected. The asynchronous port tests are halted if there is a very high loopback error rate. Ethernet port tests are halted if LAN activity is detected. After a test has halted the resource is returned to its pre-test configuration.

With the exception of the asynchronous ports, tests should not be used to test the interface through which access was obtained to the router. The reason for this is that the connection to the router will be broken when the Test Facility attaches to the interface.



No mechanism is provided to prevent this from occurring. It is the responsibility of the user to check the operation of a resource before starting the test.

Interface tests use data loopbacks and (where applicable) control line loopbacks. Frames containing a known data sequence are repeatedly transmitted via the controller being tested. The contents of frames received via the controller are compared against this sequence. To allow shorts between interfaces to be detected the transmitted sequence is unique for each interface. If a packet is received with the wrong sequence it is counted as a bad frame.

The loopback error free rate is calculated as:

$$\text{Error free} = \# \text{ good frames received} / \# \text{ frames sent}$$

where a good frame is one in which the received and transmitted data and lengths match. For ports which do not transport frames (for example, asynchronous ports), the term “frame” means the test string.

Tests can not be enabled on asynchronous ports if they are already configured for use by other modules; they must first be de-configured. The Ethernet port does not have this requirement. When tests are enabled on an Ethernet port the configurations of all the attached modules are stored and their configuration is replaced by the Test Facility.

All ports can be tested simultaneously, including the asynchronous port used to enter the test command, by entering the command:

```
ENABLE TEST INTERFACE=ALL
```

on an asynchronous port. The Test Facility detects that a test is required on an asynchronous port which was the source of the test command and only tests its control signals; the testing of the data path is made visually—if the command interpretation by the router and the response displayed on the terminal are correct then the data path is judged to be functional. A special cable is used in this case to provide a normal data path while looping the control signals (Figure 9-1 on page 9-3, Figure 9-2 on page 9-4 and Figure 9-3 on page 9-4).

Figure 9-1: Pin wiring diagram for a cable to connect a terminal to an asynchronous port that is to be tested by the Test Facility.

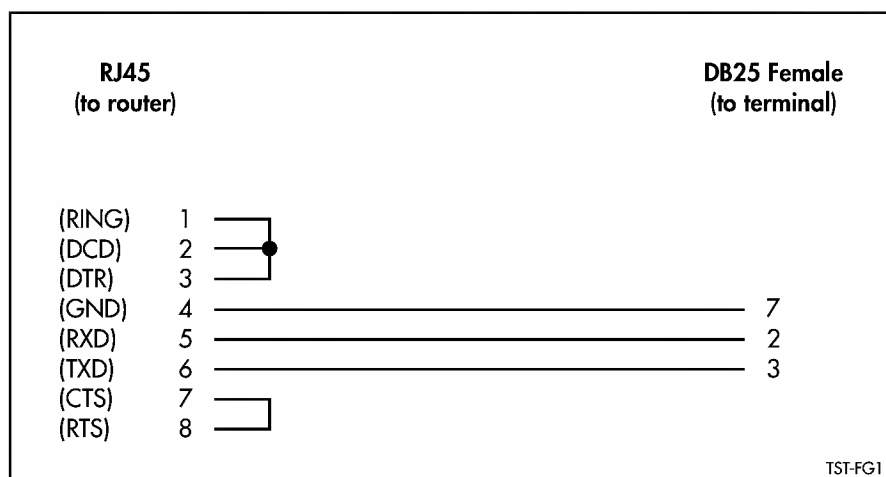


Figure 9-2: Pin wiring diagram for a cable to connect a terminal to a DB9 female asynchronous port that is to be tested by the Test Facility.

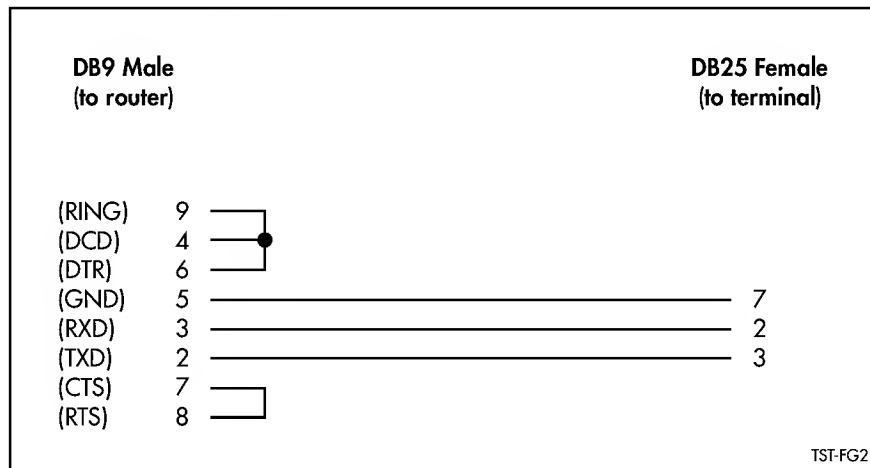
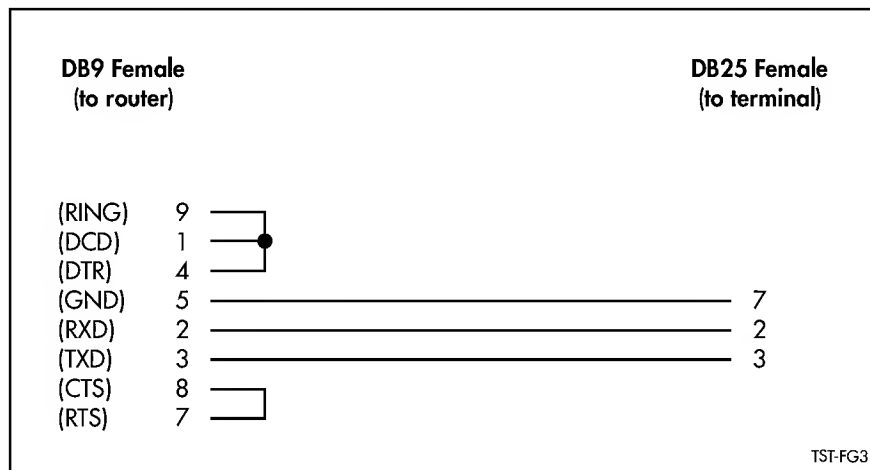


Figure 9-3: Pin wiring diagram for a cable to connect a terminal to a DB9 male asynchronous port that is to be tested by the Test Facility.



Ethernet Port Tests

The Ethernet port on the router consists of an Attachment Unit Interface (AUI) connector and/or a twisted pair (TP) connector. Two internal loopbacks (ENDEC and MAC) and one or two external loopbacks are used to test the Ethernet port. The test cycles through each loopback in turn.

To quickly detect if the test is being run on an active LAN the transceiver loopback test is run first, if data is detected on the LAN then it is assumed to be active and the test is immediately aborted. The AUI external loopback can be provided using a standard thin wire transceiver with the coax port connected to an isolated segment. Alternatively a transceiver loopback plug can be used (Figure 9-4 on page 9-5). The TP external loopback can be provided using a transceiver loopback plug (Figure 9-5 on page 9-5). The possible test outcomes are listed in Table 9-1 on page 9-5.

Figure 9-4: Ethernet AUI loopback plug wiring diagram.

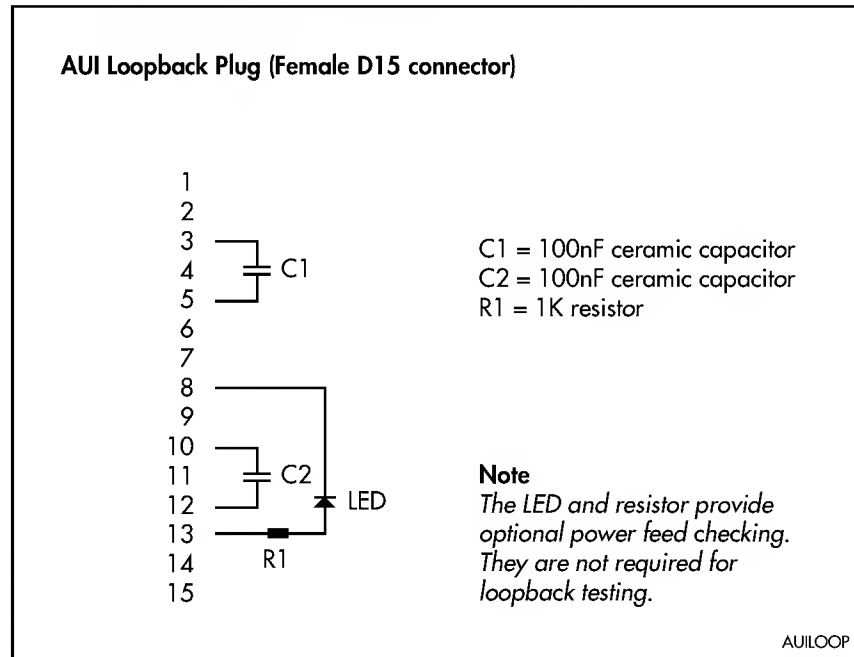


Figure 9-5: Ethernet twisted pair (TP) loopback plug wiring diagram.

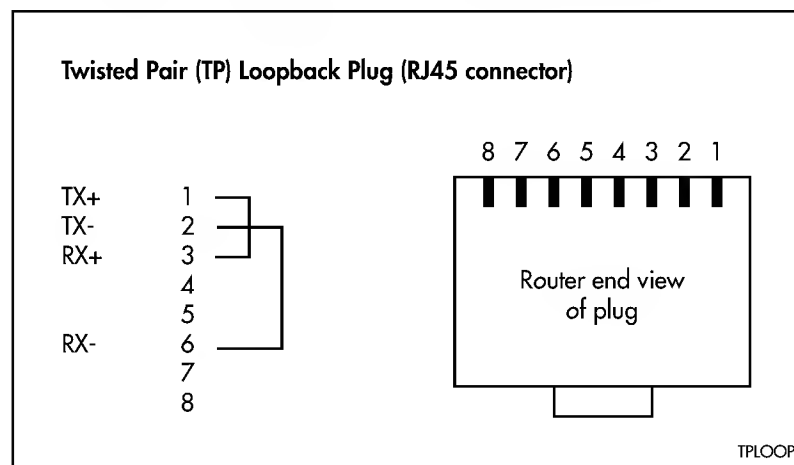


Table 9-1: Possible test outcomes for an Ethernet interface.

Event	Action	Error	Result
2 non-sent frames received in any second	Halt test	Active LAN	Bad
10 consecutive bad or missing frames during transceiver loop	Complete test	No Transceiver warning	See below
< 99.9% error free frames	Complete test	-	Bad
>= 99.9% error free frames	Complete test	-	Good

Asynchronous Port Tests

The asynchronous port test requires a loopback plug in the port being tested, to loop data and control signals back to the router (Figure 9-6 on page 9-6, Figure 9-7 on page 9-6 and Figure 9-8 on page 9-6).

Figure 9-6: RJ45 loopback plug wiring diagram.

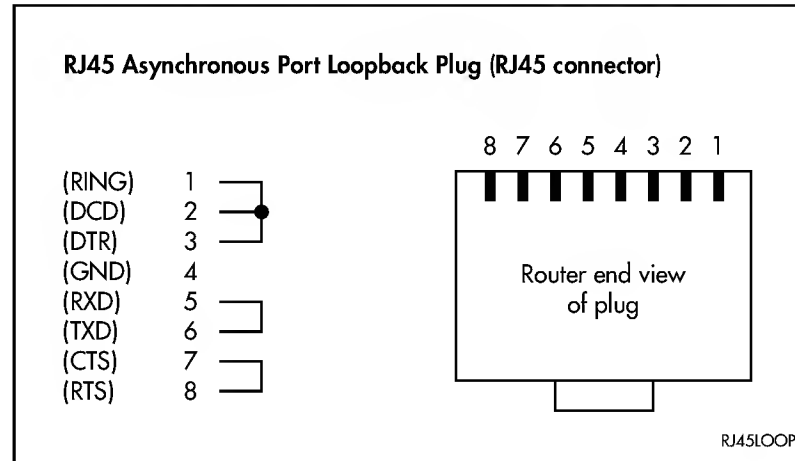


Figure 9-7: DB9 male loopback plug wiring diagram.

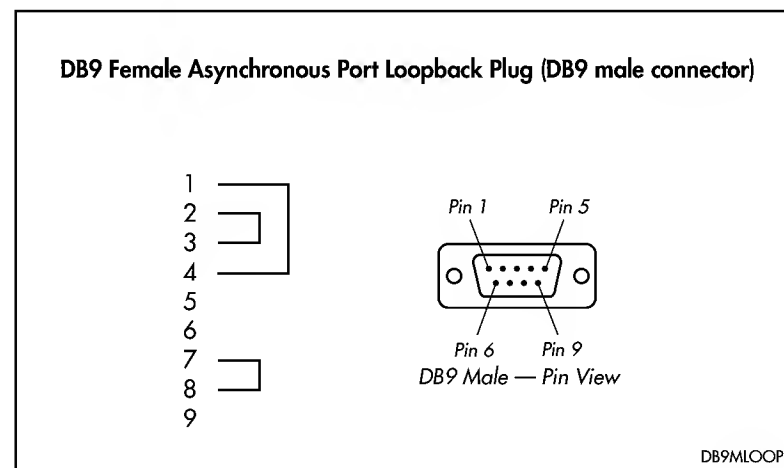
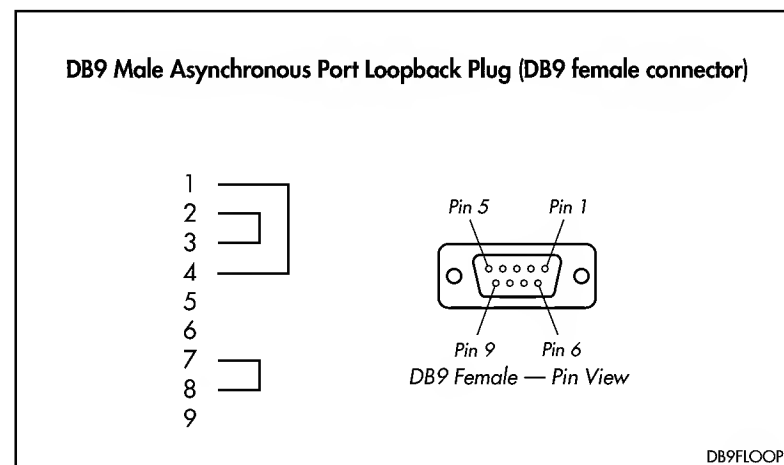


Figure 9-8: DB9 female loopback plug wiring diagram.



Three error thresholds determine the test outcome (Table 9-2 on page 9-7). For the error rate calculations a test data sequence is considered to be the equivalent of a frame.

Table 9-2: Possible test outcomes for an asynchronous interface.

Event	Action	Error	Result
10 consecutive bad or missing sequences	Halt test	No loopback	Bad
< 99.9% error free sequences	Complete test	-	Bad
>= 99.9% error free sequences	Complete test	-	Good

To test the port control signals the output signals are continuously toggled, and the corresponding (looped back) input state is examined. To pass the control signal test the state of an input must match the state of the corresponding output.



Tests can not be run on an asynchronous port which is already assigned, for example, as a Telnet session.

Basic Rate ISDN Port Tests

The Basic Rate ISDN port test requires a loopback plug in the port being tested, to loop data back to the router (Figure 9-9 on page 9-7). Three error thresholds determine the test outcome (Table 9-3 on page 9-7).

Figure 9-9: Basic Rate ISDN loopback plug wiring diagram.

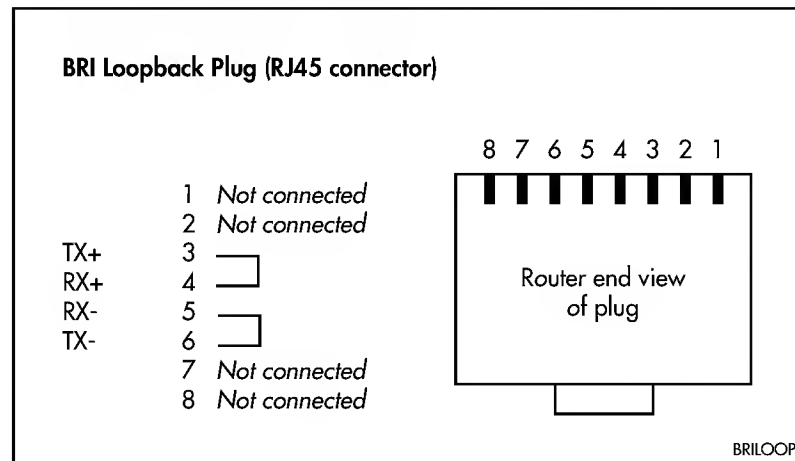


Table 9-3: Possible test outcomes for a Basic Rate ISDN interface.

Event	Action	Error	Result
10 consecutive bad or missing sequences	Halt test	No loopback	Bad
< 99.9% error free sequences	Complete test	-	Bad
>= 99.9% error free sequences	Complete test	-	Good

Tests can not be run on a BRI port which has a call established. This should not be a problem if a loopback plug is being used, because removing the ISDN connection to insert the loopback plug will disconnect the call.



It is normal for three errors to occur at the start of the test when LAPD is configured to the D channel. These errors should be ignored.

Command Reference

This section describes the commands available on the router for testing the router's hardware.

See "Conventions" on page xxxv of *Preface* in the front of this manual for details of the conventions used to describe command syntax. See *Appendix A, Messages* for a complete list of messages and their meanings.

To alert the user of a test failure the bell character is printed each time a negative test result is being printed. In the following outputs a bell character is printed for each * character displayed in the outputs (the * character is included in the actual output).

DISABLE TEST INTERFACE

Syntax `DISABLE TEST INTERFACE=interface`

where:

■ *interface* is the interface being tested.

Description This command halts the currently active interface tests. The interface or interfaces currently being tested must be specified (Table 9-4 on page 9-8).

Table 9-4: Valid interface options for the DISABLE TEST INTERFACE command.

Interface options	Tests
ALL	All router interfaces
BASE	All interfaces on the base board
ETHn	Ethernet interface n
PORTn	Asynchronous port n
BRI n	Basic Rate ISDN interface n

Examples To disable testing on Ethernet interface 0, use the command:

```
DISABLE TEST INTERFACE=ETH0
```

See Also `ENABLE TEST INTERFACE`
`RESET TEST INTERFACE`
`SHOW TEST`

ENABLE TEST INTERFACE

Syntax `ENABLE TEST INTERFACE=interface [TIME=time|CONT] [MORE]`

where:

- *interface* is the interface to be tested.
- *time* is the required test duration in minutes.

Description This command enables interface tests. The interface or interfaces to be tested must be specified (Table 9-5 on page 9-9).

Table 9-5: Valid interface options for the ENABLE TEST INTERFACE command.

Interface options	Tests
ALL	All router interfaces
BASE	All interfaces on the base board
ETHn	Ethernet interface n
PORTn	Asynchronous port n
BRIn	Basic Rate ISDN interface n

The TIME parameter specifies the duration of the tests in minutes. If TIME is not specified the tests are run for 4 minutes. If CONT is specified the tests are run continuously.

The MORE parameter provides continuous updates of the status of the current test and control states of asynchronous interfaces (Figure 9-10 on page 9-10). Control signal faults are logged to the router's logging facility, which can be displayed with the command:

```
SHOW LOG
```

The MORE parameter should only be used on a single interface at any one time. The MORE parameter is not valid when INTERFACE is set to ALL, however no mechanism is provided to prevent MORE being individually enabled on multiple interfaces. This command is provided for hardware servicing purposes only.



Due to the nature of the output it may be difficult to enter commands, including the DISABLE TEST INTERFACE command on page 9-8, while the MORE option is in effect. Tests should therefore be enabled for a specified, short period of time if the MORE option is used.

Figure 9-10: Example output from the ENABLE TEST INTERFACE MORE command for an asynchronous port.

```
port1 control signals; cycle 2

output          input
-----
rts    OFF      cts    OFF
dtr    ON       cd     ON
              ring    -
-----
```

Examples To enable testing on Ethernet interface 0, use the command:

```
ENABLE TEST INTERFACE=ETH0
```

See Also DISABLE TEST INTERFACE
RESET TEST INTERFACE
SHOW TEST

RESET TEST INTERFACE

Syntax RESET TEST INTERFACE

Description This command is used to clear all the results from interface tests, setting the state column to “no test” and clearing the other result parameters.

Examples To clear all previous test results ready to start a new test, use the command:

```
RESET TEST INTERFACE
```

See Also DISABLE TEST INTERFACE
ENABLE TEST INTERFACE
SHOW TEST

SHOW TEST

Syntax SHOW TEST [INTERFACE] [COUNTERS]

Description This command displays the unit test status and results. The results are stored until a test is rerun, the RESET TEST INTERFACE command on page 9-10 is entered, or the router is powered off or reset.

The SHOW TEST INTERFACE variant displays the results of interface tests (Figure 9-11 on page 9-11, Table 9-6 on page 9-11). The results are stored until a test is rerun, a RESET TEST INTERFACE command on page 9-10 is entered, or the router is powered off or reset.

Figure 9-11: Example output from the SHOW TEST INTERFACE command.

Board	ID	Bay	Board Name	Rev	Serial number
Base	80		AR140 (U)	M3-0	40596194
Interface	State	Result	Type	Duration (minutes)	Details Data (%OK) Control
eth0	complete	* BAD	trans	5	BAD (0.0) -
			TP	0	- - -
			ENDEC	5	good (100.0) -
			MAC	5	good (100.0) -
port0	testing	wait 12789 minutes	-	1	BAD (75.1) good .
BRI0	no test	-	-	-	- - -

Table 9-6: Parameters displayed in the output of the SHOW TEST INTERFACE command.

Parameter	Meaning
Board	The types of board installed in the router; "Base".
ID	The identification number for the board model.
Bay	<i>Not used.</i>
Board Name	The complete part name for the board.
Rev	The version number of the board.
Serial Number	The unique serial number for the board.
Interface	The name of the interface to which the test results apply.
State	The state of the test module for this interface; one of "no test", "testing", "complete" or "halted".
Result	The test result. If the test has been completed, the result will be one of "check this screen", "good" or "* BAD". If testing is in progress the result will be one of "wait continuous" or "wait <mins> minutes". If testing has been halted the result will be one of "* Active LAN", "* BAD or no SynTstr" or "* BAD or no loop".
Type	The particular test sub-mode. This varies depending on the router model and interface type being tested. Not all tests have multiple sub-modes. For example, there are four possible test sub-modes for Ethernet: "trans", "TP", "ENDEC" and "MAC".
Duration	The duration of the test.
Data	The results for data signals; one of "N/A", "good" or "BAD".
%OK	The number of data frames successfully received as a percentage of the total number of data frames transmitted.
Control	The results for control signals; one of "-", "good" or "BAD".



Due to the criteria used to halt tests the details columns of halted tests may read "good" if the event causing the test to halt occurred after the test had been running correctly.

The SHOW TEST INTERFACE COUNTERS variant displays the counters used for interface tests (Figure 9-12 on page 9-12, Table 9-7 on page 9-12). The counters are stored until a test is rerun, the RESET TEST INTERFACE command on page 9-10 is entered or the router is powered off or reset.

Figure 9-12: Example output from the SHOW TEST INTERFACE COUNTERS command.

Interface	State	Type	Duration (minutes)	Frame Counters			
				Tx	RxTotal	RxGood	RxBad
eth0	complete	trans	10	000453728	000453727	000453727	000000000
		TP	0	000000000	000000000	000000000	000000000
		ENDEC	10	000456098	000456097	000456097	000000000
		MAC	10	000549876	000549876	000549875	000000000
port0	testing	-	1	000000176	000000175	000000175	000000000
BRI0	no test	-	-	-	-	-	-

Table 9-7: Parameters displayed in the output of the SHOW TEST INTERFACE COUNTERS command.

Parameter	Meaning
Interface	The name of the interface to which the test counters apply.
State	The state of the test module for this interface; one of "no test", "testing", "complete" or "halted".
Type	The particular test sub-mode. This is test dependent and not all tests have multiple sub-modes. For example, there are four possible test sub-modes for Ethernet: "trans", "TP", "ENDEC" and "MAC".
Duration	The duration of the test.
Tx	The total number of frames transmitted on the interface.
RxTotal	The total number of frames received on the interface.
RxGood	The number of good frames received on the interface.
RxBad	The number of bad frames received on the interface.

Examples To display the test results, use the command:

```
SHOW TEST COUNTERS
```

See Also DISABLE TEST INTERFACE
ENABLE TEST INTERFACE
RESET TEST INTERFACE

Chapter 10

Trigger Facility

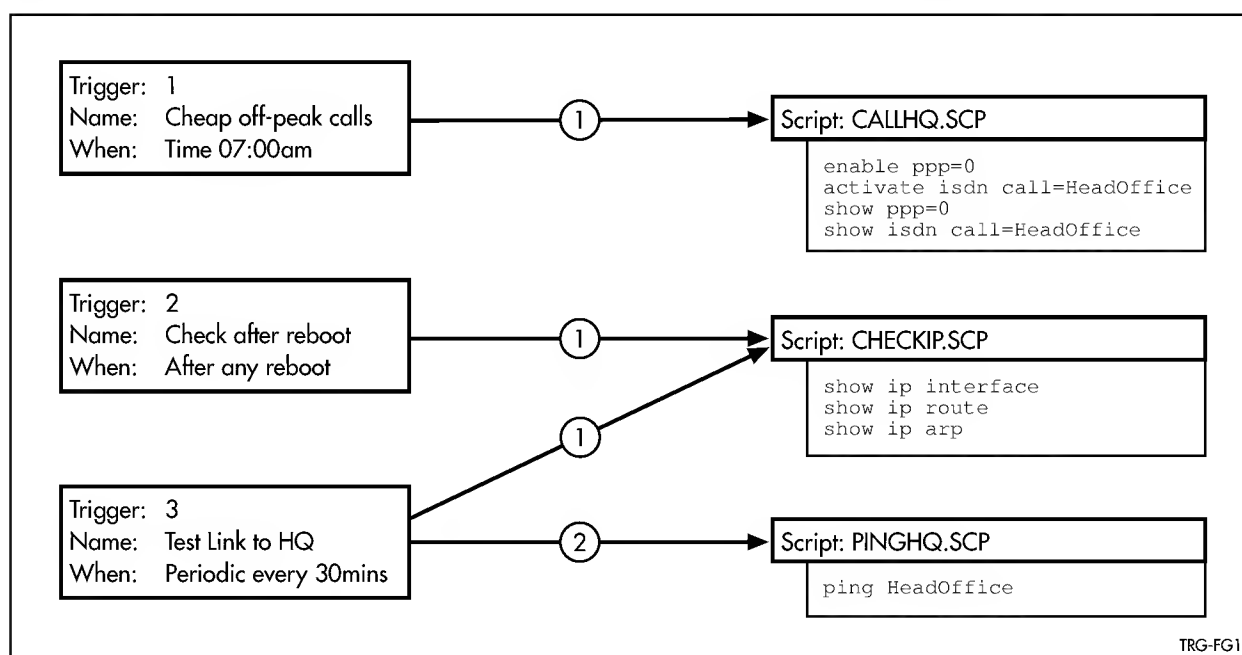
Introduction	10-2
Defining Triggers	10-3
Examples	10-3
Initiating ISDN Calls During Off-Peak Periods	10-3
Retrieving System Snapshots After a Reboot	10-5
Command Reference	10-5
ACTIVATE TRIGGER	10-5
ADD TRIGGER	10-6
CREATE TRIGGER	10-7
DELETE TRIGGER	10-11
DESTROY TRIGGER	10-11
DISABLE TRIGGER	10-12
ENABLE TRIGGER	10-12
PURGE TRIGGER	10-13
SET TRIGGER	10-13
SHOW TRIGGER	10-16

Introduction

The trigger facility provides a powerful mechanism for automatic and timed management of the router, by automating the execution of router commands in response to certain events. For example, triggers can be configured to activate and deactivate ISDN calls at specified times, or to collect diagnostic information after a router reboot.

A trigger is an ordered sequence of scripts to be executed when a certain event occurs. A script is a sequence of router commands stored as a plain text file in the router's file subsystem in FLASH memory. Each trigger may reference multiple scripts and any script may be used by any trigger. Various types of triggers are supported, each activated in a different way (Figure 10-1 on page 10-2).

Figure 10-1: Triggers respond to events by performing a sequence of predefined scripts.



When a trigger is activated, the scripts associated with it will be executed in sequence by the router. The output from the scripts is passed to the logging facility, and can be displayed with the SHOW LOG command on page 12-31 or forwarded to another router.

Defining Triggers

A trigger is created using the command:

```
CREATE TRIGGER=trigger-id type [type-parameters]
    [SCRIPT=filename] [NAME=name] [REPEAT={YES|NO|ONCE|
FOREVER|count}] [STATE={ENABLED|DISABLED}] [TEST={YES|NO|
ON|OFF}]
```

A trigger is modified using the command:

```
SET TRIGGER=trigger-id type [type-parameters]
    [SCRIPT=filename] [NAME=name] [REPEAT={YES|NO|ONCE|
FOREVER|count}] [STATE={ENABLED|DISABLED}] [TEST={YES|NO|
ON|OFF}]
```

Each trigger can be assigned a descriptive name and up to five scripts can be executed. See *Chapter 13, Scripting* for more information about creating scripts. An unlimited number of additional scripts can be added to the trigger (up to five at a time), at any position in the sequence, using the command:

```
ADD TRIGGER=trigger-id SCRIPT=filename... [NUMBER=index]
```

Scripts can be deleted from a trigger using the command:

```
DELETE TRIGGER=trigger-id NUMBER=index
```

A trigger can be destroyed using the command:

```
DESTROY TRIGGER=trigger-id
```

By default, triggers are enabled when created. A trigger can be enabled or disabled, using the commands:

```
ENABLE TRIGGER=trigger-id
DISABLE TRIGGER=trigger-id
```

or explicitly activated (triggered), using the command:

```
ACTIVATE TRIGGER=trigger-id
```

regardless of whether or not the trigger is currently enabled. The command:

```
SHOW TRIGGER=trigger-id [{FULL|SUMMARY|STATUS|COUNT}]
```

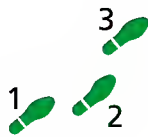
displays summary information about all defined triggers or detailed information about a particular trigger.

Examples

The following examples illustrate some of the possible uses for the trigger facility.

Initiating ISDN Calls During Off-Peak Periods

Suppose that the charging regime for regional ISDN calls is such that calls initiated between 8am and 8pm are charged at a rate twice that of calls initiated between 8pm and 8am, and that the charging rate for the entire call is based on the charging rate at the time the call is initiated, regardless of how long the call lasts. For a network link that is busy often during the working day, it may be cheaper to activate the call before 8am and keep the link up until 6pm, than to make numerous calls after 8am.



To automatically activate ISDN calls during off-peak charging periods:

1. Create the ISDN call.

Create the ISDN call, specifying any required options, using the command:

```
ADD ISDN CALL=CHEAP NUMBER=42 PRECEDENCE=OUT
```

2. Create a PPP interface to use the ISDN call.

Create a PPP interface to use the call, and set the idle time to the default of 36000 seconds (the number of seconds between 8am and 6pm). This ensures that the call will stay up once it is activated at 8am.

```
CREATE PPP=0 OVER=ISDN-CHEAP IDLE=36000
```

3. Create an script to activate the ISDN call.

Create a script that explicitly sets the idle time of the PPP interface to 36000 seconds (to keep the call up all day), and activates the ISDN call:

```
ADD SCRIPT=ACHEAP.SCP TEXT="SET PPP=0 IDLE=36000"
ADD SCRIPT=ACHEAP.SCP TEXT="ACTIVATE ISDN CALL=CHEAP"
```

4. Create a trigger to use the script.

Create a time trigger to activate at 7:59am and execute the script:

```
ENABLE TRIGGER
CREATE TRIGGER=1 TIME=07:59 DAYS=WEEKDAYS
SCRIPT=ACHEAP.SCP NAME="Enable Off-Peak Calls"
REPEAT=FOREVER
```

5. Create a script to deactivate the ISDN call.

Create a script that explicitly sets the idle time of the PPP interface to the default of 60 seconds (to activate the call when there is traffic during the night), and deactivates the ISDN call:

```
ADD SCRIPT=DCHEAP.SCP TEXT="SET PPP=0 IDLE=60"
```

Note that the ISDN call is not explicitly deactivated, in case there is traffic being transmitted over the link. The call will automatically deactivate when there has been no traffic for 60 seconds.

6. Create a trigger to use the script.

Create a time trigger to activate at 6pm and execute the script:

```
CREATE TRIGGER=2 TIME=18:00 DAYS=WEEKDAYS
SCRIPT=DCHEAP.SCP NAME="Disable Off-Peak Calls"
REPEAT=FOREVER
```

7. Save the dynamic configuration

Save the modified dynamic configuration as the script file OFFPEAK.CFG and make OFFPEAK.CFG the boot script:

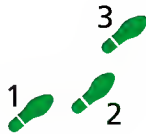
```
SAVE CONFIG=OFFPEAK.CFG
SET CONFIG=OFFPEAK.CFG
```

Retrieving System Snapshots After a Reboot

When the router reboots (due to a RESTART command or a fatal error), information about the state of the router system before the reboot can be obtained from the output of the command:

```
SHOW DEBUG
```

However, this information is lost (replaced) the next time the router reboots. To ensure that information about every router reboot is retained, a trigger can be created to take a snapshot of the system after every reboot.



To automatically generate a system snapshot after a router reboot:

1. Create a script to generate the snapshot.

Create a script that executes the SHOW DEBUG command:

```
ADD SCRIPT=SNAPSHOT.SCP TEXT="SHOW DEBUG"
```

2. Create a trigger to use the script.

Create a reboot trigger to execute the script after every router reboot:

```
ENABLE TRIGGER
CREATE TRIGGER=1 REBOOT=ALL DAYS=ALL SCRIPT=SNAPSHOT.SCP
NAME="Reboot" REPEAT=FOREVER
```

3. Save the dynamic configuration

Save the modified dynamic configuration as the script file SNAPSHOT.CFG and make SNAPSHOT.CFG the boot script:

```
SAVE CONFIG=SNAPSHOT.CFG
SET CONFIG=SNAPSHOT.CFG
```

Command Reference

This section describes the commands to configure and manage the trigger facility in the router. The trigger facility requires that the router's internal clock be set correctly. See *Chapter 1, Operation* for detailed descriptions of the commands required to configure the router's internal clock.

See "Conventions" on page xxxv of *Preface* in the front of this manual for details of the conventions used to describe command syntax. See *Appendix A, Messages* for a complete list of messages and their meanings.

ACTIVATE TRIGGER

Syntax ACTIVATE TRIGGER=*trigger-id*

where:

- *trigger-id* is a number in the range 1 to 100.

Description This command manually and immediately activates the specified trigger, even if it has been disabled using the DISABLE TRIGGER command on page 10-12. The scripts associated with the trigger will be executed even if the TEST option

is set. Normally, a TEST trigger generates only log entries when it triggers and does not cause any scripts to be invoked.

The TRIGGER parameter specifies the number of the trigger to activate. The specified trigger must already exist.

Triggers activated manually do not have their repeat counts decremented or their *"last triggered"* time updated, and do not result in updates to the *"time/periodic triggers today"* counters.

Examples To activate trigger number 8, use the command:

```
ACTIVATE TRIGGER=8
```

See Also CREATE TRIGGER
DISABLE TRIGGER
ENABLE TRIGGER
SHOW TRIGGER

ADD TRIGGER

Syntax ADD TRIGGER=*trigger-id* SCRIPT=*filename...* [NUMBER=*index*]

where:

- *trigger-id* is a number in the range 1 to 100.
- *filename* is a file name of the form `device:filename.type`. *device* is the name of the memory device in which the file is stored (e.g. FLASH, NVS). *type* must be "SCP" or "CFG". Valid characters are uppercase letters (A–Z), lowercase letters (a–z), digits (0–9) and the hyphen character (-). Wildcards are not allowed.
- *index* is a number in the range 1 to *n*+1, where *n* is the number of scripts already assigned to the trigger.

Description This command adds a script to a trigger, so that the script will be executed when the trigger is activated.

The TRIGGER parameter specifies the number of the trigger to which the script is to be added. The specified trigger must already exist.

The SCRIPT parameter specifies the name of the script to be added. The SCRIPT parameter may be repeated up to five times in one command, to add up to five scripts (in the order specified) at once.

The NUMBER parameter specifies the position in the script list where the script is to be added. If NUMBER is specified, the new script will occupy that position and all following scripts will be pushed down one position.

Examples To add scripts SNAPSHOT.SCP and CALLHQ.SCP to trigger 1 at position 3, use the command:

```
ADD TRIGGER=1 SCRIPT=SNAPSHOT.SCP SCRIPT=CALLHQ.SCP NUMBER=3
```

See Also CREATE TRIGGER
DELETE TRIGGER
DISABLE TRIGGER
ENABLE TRIGGER
SET TRIGGER
SHOW TRIGGER

CREATE TRIGGER

Syntax CREATE TRIGGER=*trigger-id* [CPU=*value* [DIRECTION={UP|DOWN|ANY}]] [INTERFACE=*interface* EVENT={UP|DOWN|FAIL|ANY} [CIRCUIT=*miox-circuit*] [CP={BCP|CCP|IPCP|LCP}] [MEMORY=*value* [DIRECTION={UP|DOWN|ANY}]] [PERIODIC=*minutes*] [REBOOT={RESTART|CRASH|ALL}] [TIME=*hh:mm*] [DATE=*date*] [DAYS=*day-list*] [AFTER=*hh:mm*] [BEFORE=*hh:mm*] [SCRIPT=*filename...*] [NAME=*name*] [REPEAT={YES|NO|ONCE|FOREVER|*count*}] [STATE={ENABLED|DISABLED}] [TEST={YES|NO|ON|OFF}]

where:

- *trigger-id* is a number in the range 1 to 100.
- *value* is a number in the range 1 to 100.
- *interface* is an interface name formed by concatenating an interface type and an interface instance (e.g. eth0).
- *miox-circuit* is an alphanumeric string, 1 to 15 characters in length.
- *minutes* is a number in the range 1 to 1439.
- *hh:mm* is a time in hours and minutes.
- *date* is a date in the format dd-mmm-yyyy, where *mmm* is the first three letters of the month name.
- *day-list* is one or more of the keywords "MON", "TUE", "WED", "THU", "FRI", "SAT", "SUN", "WEEKDAY", "WEEKEND" or "ALL", separated by commas.
- *filename* is a file name of the form `device:filename.type`. *device* is the name of the memory device in which the file is stored (e.g. FLASH, NVS). *type* must be "SCP" or "CFG". Valid characters are uppercase letters (A–Z), lowercase letters (a–z), digits (0–9) and the hyphen character (-). Wildcards are not allowed.
- *name* is a character string, 1 to 40 characters in length. If the string contains spaces it must be enclosed in double quotes.
- *count* is a number in the range 1 to 4294967294 ($2^{32}-2$).

Description This command creates a new trigger. Seven different trigger types are supported—CPU triggers, link triggers, memory triggers, periodic triggers, reboot triggers and time triggers. Some parameters are specific to a particular trigger type, while others are applicable to more than one trigger type.

The TRIGGER parameter specifies the number of the trigger to create. The number is used to reference the trigger in other commands. The specified trigger must not already exist. The TRIGGER parameter must immediately

follow the CREATE keyword, and must be followed immediately by one of the parameters TIME, REBOOT, PERIODIC, INTERFACE, CPU or MEMORY.

The CPU parameter defines a CPU utilisation trigger and specifies the CPU utilisation level at which the trigger is to be activated. The type-specific parameters DIRECTION, DATE or DAYS, AFTER and BEFORE, and the general parameters SCRIPT, NAME, REPEAT, STATE and TEST may also be specified.

The DIRECTION parameter specifies how the CPU or memory utilisation threshold is reached to activate the trigger. If UP is specified, the trigger is activated when CPU or memory utilisation increases to or exceeds the threshold. If DOWN is specified, the trigger is activated when CPU or memory utilisation decreases to or falls below the threshold. If ANY is specified, the trigger is activated when CPU or memory utilisation equals or passes the threshold in either direction. The default is ANY.

The INTERFACE parameter defines an interface (link) trigger and specifies the interface to monitor. The INTERFACE parameter must be followed by the EVENT parameter. The type-specific parameters CIRCUIT and CP, and the general parameters SCRIPT, NAME, REPEAT, STATE and TEST may also be specified.

The EVENT parameter specifies a link (interface) status change event (up, down, failed to open or any of these) and must follow the INTERFACE parameter. It is not valid with any other trigger type.

The CIRCUIT parameter specifies a MIOX circuit to monitor. The trigger will only activate if the condition defined by the combination of INTERFACE, CIRCUIT and EVENT occurs. The CIRCUIT parameter may only follow an INTERFACE parameter that specifies an X25T interface. When activated, the trigger will pass three parameters to the trigger scripts(s)—the X.25 instance that caused the trigger, the name of the MIOX circuit and the event state.

The CP parameter specifies a PPP control protocol to monitor. The trigger will only activate if the condition defined by the combination of INTERFACE, CP and EVENT occurs. The CP parameter may only follow an INTERFACE parameter that specifies a PPP interface. When activated, the trigger will pass three parameters to the trigger scripts(s)—the PPP instance that caused the trigger, the control protocol and the event state.

The MEMORY parameter defines a memory utilisation trigger and specifies the amount of free memory at which the trigger is to be activated. The type-specific parameters DIRECTION, DATE or DAYS, AFTER and BEFORE, and the general parameters SCRIPT, NAME, REPEAT, STATE and TEST may also be specified.

The PERIODIC parameter defines a periodic trigger and specifies the period of the trigger in minutes. The type-specific parameters DATE or DAYS, AFTER and BEFORE, and the general parameters SCRIPT, NAME, REPEAT, STATE and TEST may also be specified.

The REBOOT parameter defines a reboot trigger and specifies a list of reboot events that will activate the trigger. If CRASH is specified, the trigger will be activated by a router crash. If RESTART is specified, the trigger will be activated by any reboot other than a router crash. If ALL is specified, the trigger will be activated by any reboot event. The type-specific parameters DATE or DAYS, AFTER and BEFORE, and the general parameters SCRIPT, NAME, REPEAT, STATE and TEST may also be specified.

The **TIME** parameter defines a time trigger and specifies the time of day in hours and minutes when the trigger is to be activated. Resolutions of up to one minute with an accuracy of five seconds are supported. The trigger will activate at most five seconds after the specified minute. The type-specific parameters **DATE** or **DAYS**, and the general parameters **SCRIPT**, **NAME**, **REPEAT**, **STATE** and **TEST** may also be specified.

The **AFTER** parameter specifies the earliest time of day that the trigger will activate, in hours and minutes. The trigger may activate any time between the time specified and midnight.

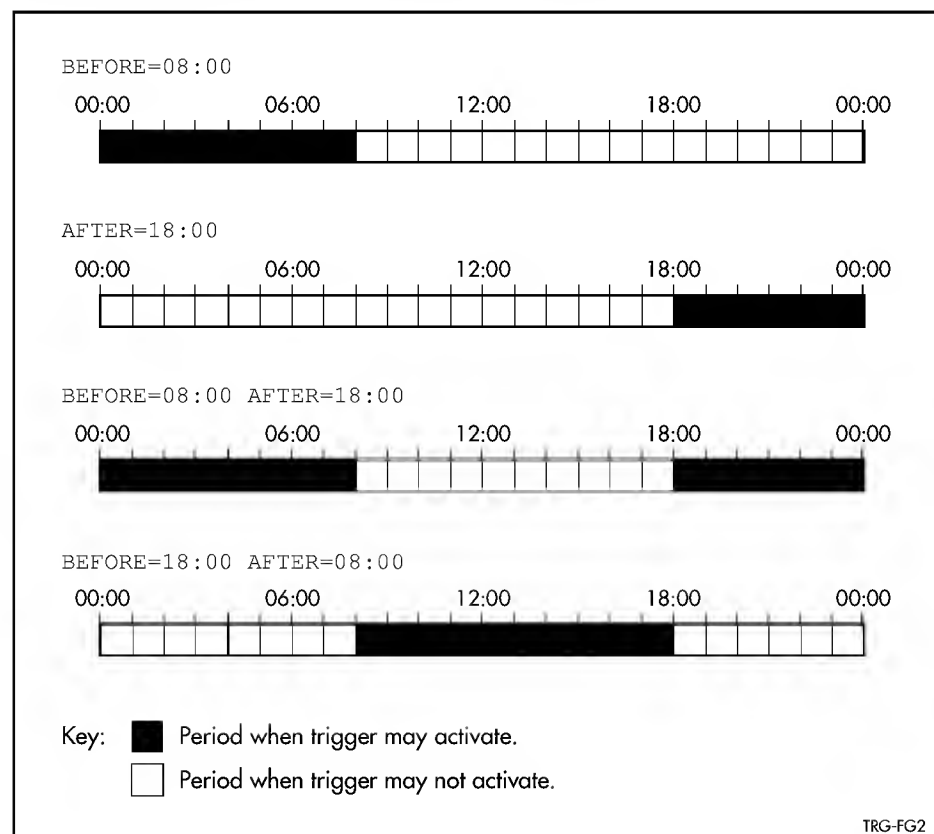
The **BEFORE** parameter specifies the latest time of day that the trigger will activate, in hours and minutes. The trigger may activate any time between midnight and the time specified.

If neither **AFTER** nor **BEFORE** are specified, there is no restriction on when the trigger may activate. If both **AFTER** and **BEFORE** are specified, and the time periods specified overlap, the trigger may activate any time during the overlap period (Figure 10-2 on page 10-9).

The **DATE** parameter specifies a date on which the trigger may activate. The **DAYS** and **DATE** parameters are mutually exclusive—use one or the other.

The **DAYS** parameter specifies a list of days, separated by commas, on which the trigger will activate. The value **WEEKDAY** is a synonym for the list of values "MON, TUE, WED, THU, FRI" and **WEEKEND** is a synonym for the list of values "SAT, SUN". Any combination of these and the day names is acceptable. The default is **ALL**. The **DAYS** and **DATE** parameters are mutually exclusive—use one or the other.

Figure 10-2: The effects of different combinations of the **AFTER and **BEFORE** parameters in the **CREATE TRIGGER** and **SET TRIGGER** commands.**



The NAME parameter specifies a descriptive name for this trigger.

The REPEAT parameter specifies whether or not the trigger will repeat, or a count of how many times the trigger will repeat. If YES or FOREVER is specified, the repeat count is set to a value that is effectively infinite. If NO or ONCE is specified, the trigger will activate only once. If a numeric value is specified, the trigger will repeat the specified number of times. The default is FOREVER.

The SCRIPT parameter specifies the name of a script to execute when the trigger is activated. A script is a predefined list of router commands. The specified script must already exist. The SCRIPT parameter may be repeated up to five times in one command, to add up to five scripts (in the order specified) at once. Additional scripts may be added using the ADD TRIGGER command on page 10-6.

The STATE parameter specifies the initial state of the trigger. By default triggers are enabled when created. A trigger will only activate automatically if it is enabled. A trigger can be manually activated with the ACTIVATE TRIGGER command on page 10-5 regardless of whether the trigger is enabled or disabled.

The TEST parameter specifies whether or not this trigger is in TEST mode. When a trigger is in test mode, it activates and logs the trigger, but does not execute any configured scripts. The default is NO.

Examples To create trigger 1 that activates at 6am every weekday and initiates script OFFPEAK.SCP, use the command:

```
CREATE TRIGGER=1 TIME=06:00 DAYS=WEEKDAY SCRIPT=OFFPEAK.SCP
REPEAT=YES
```

To create trigger 3 that executes script IPCALLG.SCP when IPCP closes on interface ppp3, use the command:

```
CREATE TRIGGER=3 INTERFACE=PPP3 EVENT=DOWN CP=IPCP
SCRIPT=IPCALLG.SCP
```

See Also ACTIVATE TRIGGER
ADD TRIGGER
DESTROY TRIGGER
DISABLE TRIGGER
ENABLE TRIGGER
SET TRIGGER
SHOW TRIGGER

DELETE TRIGGER

Syntax DELETE TRIGGER=*trigger-id* NUMBER=*index*

where:

- *trigger-id* is a number in the range 1 to 100.
- *index* is a number in the range 1 to *n*, where *n* is the number of scripts assigned to the trigger.

Description This command removes a script from a trigger. The TRIGGER parameter specifies the number of the trigger from which the script is to be deleted. The specified trigger must already exist.

The NUMBER parameter specifies the position in the script list of the script to be removed. The specified script must already exist in that position.

Examples To remove the third script from trigger 1, use the command:

```
DELETE TRIGGER=1 NUMBER=3
```

See Also ADD TRIGGER
DESTROY TRIGGER
SET TRIGGER
SHOW TRIGGER

DESTROY TRIGGER

Syntax DESTROY TRIGGER=*trigger-id*

where:

- *trigger-id* is a number in the range 1 to 100.

Description This command destroys a previously-defined trigger. The TRIGGER parameter specifies the number of the trigger to destroy. The specified trigger must already exist.

Examples To destroy trigger 1, use the command:

```
DESTROY TRIGGER=1
```

See Also ADD TRIGGER
CREATE TRIGGER
DELETE TRIGGER
DISABLE TRIGGER
ENABLE TRIGGER
PURGE TRIGGER
SHOW TRIGGER

DISABLE TRIGGER

Syntax `DISABLE TRIGGER[=trigger-id]`

where:

- *trigger-id* is a number in the range 1 to 100.

Description This command disables the entire trigger facility, if a trigger is not specified, or the specified trigger. The specified trigger is no longer eligible for activation. The trigger may still be manually activated using the `ACTIVATE TRIGGER` command on page 10-5.

The `TRIGGER` parameter specifies the number of the trigger to disable. The specified trigger must already exist.

Examples To disable trigger 1, use the command:

```
DISABLE TRIGGER=1
```

To disable the trigger module, use the command:

```
DISABLE TRIGGER
```

See Also `ACTIVATE TRIGGER`
`DELETE TRIGGER`
`DESTROY TRIGGER`
`ENABLE TRIGGER`
`PURGE TRIGGER`
`SHOW TRIGGER`

ENABLE TRIGGER

Syntax `ENABLE TRIGGER[=trigger-id]`

where:

- *trigger-id* is a number in the range 1 to 100.

Description This command enables the entire trigger facility, if a trigger is not specified, or the specified trigger. The specified trigger is eligible for activation. All triggers are enabled by default when they are created. Except for manual activation (using the `ACTIVATE TRIGGER` command on page 10-5) disabled triggers can not be activated.

The `TRIGGER` parameter specifies the number of the trigger to enable. The specified trigger must already exist.

Examples To enable trigger 1, use the command:

```
ENABLE TRIGGER=1
```

To enable the trigger module, use the command:

```
ENABLE TRIGGER
```

See Also ACTIVATE TRIGGER
 DELETE TRIGGER
 DESTROY TRIGGER
 DISABLE TRIGGER
 PURGE TRIGGER
 SET TRIGGER
 SHOW TRIGGER

PURGE TRIGGER

Syntax PURGE TRIGGER

Description This command erases the trigger facility configuration.

See Also DELETE TRIGGER
 DESTROY TRIGGER
 DISABLE TRIGGER
 ENABLE TRIGGER
 SET TRIGGER
 SHOW TRIGGER

SET TRIGGER

Syntax SET TRIGGER=*trigger-id* [CPU=*value* [DIRECTION={UP|DOWN|ANY}]] [INTERFACE=*interface* EVENT={UP|DOWN|FAIL|ANY} [CIRCUIT=*miox-circuit*] [CP={BCP|CCP|IPCP|LCP}] [MEMORY=*value* [DIRECTION={UP|DOWN|ANY}]] [PERIODIC=*minutes*] [REBOOT={RESTART|CRASH|ALL}] [TIME=*hh:mm*] [DATE=*date*] [DAYS=*day-list*] [AFTER=*hh:mm*] [BEFORE=*hh:mm*] [SCRIPT=*filename...*] [NAME=*name*] [REPEAT={YES|NO|ONCE|FOREVER|*count*}] [STATE={ENABLED|DISABLED}] [TEST={YES|NO|ON|OFF}]

where:

- *trigger-id* is a number in the range 1 to 100.
- *value* is a number in the range 1 to 100.
- *interface* is an interface name formed by concatenating an interface type and an interface instance (e.g. eth0).
- *miox-circuit* is an alphanumeric string, 1 to 15 characters in length.
- *minutes* is a number in the range 1 to 1439.
- *hh:mm* is a time in hours and minutes.
- *date* is a date in the format dd-mmm-yyyy, where *mmm* is the first three letters of the month name.
- *day-list* is one or more of the keywords "MON", "TUE", "WED", "THU", "FRI", "SAT", "SUN", "WEEKDAY", "WEEKEND" or "ALL", separated by commas.

- *filename* is a file name of the form `device:filename.type`. *device* is the name of the memory device in which the file is stored (e.g. FLASH, NVS). *type* must be "SCP" or "CFG". Valid characters are uppercase letters (A–Z), lowercase letters (a–z), digits (0–9) and the hyphen character (-). Wildcards are not allowed.
- *name* is a character string, 1 to 40 characters in length. If the string contains spaces it must be enclosed in double quotes.
- *count* is a number in the range 1 to 4294967294 ($2^{32}-2$).

Description This command modifies the definition of a trigger. Seven different trigger types are supported—CPU triggers, link triggers, memory triggers, periodic triggers, reboot triggers and time triggers. Some parameters are specific to a particular trigger type, while others are applicable to more than one trigger type.

The CPU parameter defines a CPU utilisation trigger and specifies the CPU utilisation level at which the trigger is to be activated. The type-specific parameters DIRECTION, DATE or DAYS, AFTER and BEFORE, and the general parameters SCRIPT, NAME, REPEAT, STATE and TEST may also be specified.

The DIRECTION parameter specifies how the CPU or memory utilisation threshold is reached to activate the trigger. If UP is specified, the trigger is activated when CPU or memory utilisation increases to or exceeds the threshold. If DOWN is specified, the trigger is activated when CPU or memory utilisation decreases to or falls below the threshold. If ANY is specified, the trigger is activated when CPU or memory utilisation equals or passes the threshold in either direction. The default is ANY.

The INTERFACE parameter defines an interface (link) trigger and specifies the interface to monitor. The INTERFACE parameter must be followed by the EVENT parameter. The type-specific parameters CIRCUIT and CP, and the general parameters SCRIPT, NAME, REPEAT, STATE and TEST may also be specified.

The EVENT parameter specifies a link (interface) status change event (up, down, failed to open or any of these) and must follow the INTERFACE parameter. It is not valid with any other trigger type.

The CIRCUIT parameter specifies a MIOX circuit to monitor. The trigger will only activate if the condition defined by the combination of INTERFACE, CIRCUIT and EVENT occurs. The CIRCUIT parameter may only follow an INTERFACE parameter that specifies an X25T interface. When activated, the trigger will pass three parameters to the trigger scripts(s)—the X.25 instance that caused the trigger, the name of the MIOX circuit and the event state.

The CP parameter specifies a PPP control protocol to monitor. The trigger will only activate if the condition defined by the combination of INTERFACE, CP and EVENT occurs. The CP parameter may only follow an INTERFACE parameter that specifies a PPP interface. When activated, the trigger will pass three parameters to the trigger scripts(s)—the PPP instance that caused the trigger, the control protocol and the event state.

The MEMORY parameter defines a memory utilisation trigger and specifies the amount of free memory at which the trigger is to be activated. The type-specific parameters DIRECTION, DATE or DAYS, AFTER and BEFORE, and the general parameters SCRIPT, NAME, REPEAT, STATE and TEST may also be specified.

The PERIODIC parameter defines a periodic trigger and specifies the period of the trigger in minutes. The type-specific parameters DATE or DAYS, AFTER and BEFORE, and the general parameters SCRIPT, NAME, REPEAT, STATE and TEST may also be specified.

The REBOOT parameter defines a reboot trigger and specifies a list of reboot events that will activate the trigger. If CRASH is specified, the trigger will be activated by a router crash. If RESTART is specified, the trigger will be activated by any reboot other than a router crash. If ALL is specified, the trigger will be activated by any reboot event. The type-specific parameters DATE or DAYS, AFTER and BEFORE, and the general parameters SCRIPT, NAME, REPEAT, STATE and TEST may also be specified.

The TIME parameter defines a time trigger and specifies the time of day in hours and minutes when the trigger is to be activated. Resolutions of up to one minute with an accuracy of five seconds are supported. The trigger will activate at most five seconds after the specified minute. The type-specific parameters DATE or DAYS, and the general parameters SCRIPT, NAME, REPEAT, STATE and TEST may also be specified.

The AFTER parameter specifies the earliest time of day that the trigger will activate, in hours and minutes. The trigger may activate any time between the time specified and midnight.

The BEFORE parameter specifies the latest time of day that the trigger will activate, in hours and minutes. The trigger may activate any time between midnight and the time specified.

If neither AFTER nor BEFORE are specified, there is no restriction on when the trigger may activate. If both AFTER and BEFORE are specified, and the time periods specified overlap, the trigger may activate any time during the overlap period (Figure 10-2 on page 10-9).

The DATE parameter specifies a date on which the trigger may activate. The DAYS and DATE parameters are mutually exclusive—use one or the other.

The DAYS parameter specifies a list of days, separated by commas, on which the trigger will activate. The value WEEKDAY is a synonym for the list of values "MON, TUE, WED, THU, FRI" and WEEKEND is a synonym for the list of values "SAT, SUN". Any combination of these and the day names is acceptable. The default is ALL. The DAYS and DATE parameters are mutually exclusive—use one or the other.

The NAME parameter specifies a descriptive name for this trigger.

The REPEAT parameter specifies whether or not the trigger will repeat, or a count of how many times the trigger will repeat. If YES or FOREVER is specified, the repeat count is set to a value that is effectively infinite. If NO or ONCE is specified, the trigger will activate only once. If a numeric value is specified, the trigger will repeat the specified number of times. The default is FOREVER.

The SCRIPT parameter specifies the name of a script to execute when the trigger is activated. A script is a predefined list of router commands. The specified script must already exist. The SCRIPT parameter may be repeated up to five times in one command, to add up to five scripts (in the order specified) at once. Additional scripts may be added using the ADD TRIGGER command on page 10-6.

The STATE parameter specifies the initial state of the trigger. By default triggers are enabled when created. A trigger will only activate automatically if it is enabled. A trigger can be manually activated with the ACTIVATE TRIGGER command on page 10-5 regardless of whether the trigger is enabled or disabled.

The TEST parameter specifies whether or not this trigger is in TEST mode. When a trigger is in test mode, it activates and logs the trigger, but does not execute any configured scripts. The default is NO.

Examples To modify trigger 1 to activate at 8am every weekday, use the command:

```
SET TRIGGER=1 TIME=08:00 DAYS=WEEKDAY REPEAT=YES
```

See Also ACTIVATE TRIGGER
ADD TRIGGER
CREATE TRIGGER
DESTROY TRIGGER
DISABLE TRIGGER
ENABLE TRIGGER
SET TRIGGER

SHOW TRIGGER

Syntax SHOW TRIGGER[=*trigger-id*] [{COUNTERS|FULL|STATUS|SUMMARY}]

where:

■ *trigger-id* is a number in the range 1 to 100.

Description This command displays information about all triggers that have been configured, a specified trigger, or general configuration information for the trigger facility. The TRIGGER parameter specifies the number of the trigger to display. The specified trigger must already exist.

If no trigger or parameter is specified, or the SUMMARY parameter is specified, summary information for all or the specified triggers is displayed (Figure 10-3 on page 10-17, Table 10-1 on page 10-17). If FULL is specified, or a trigger is specified without the SUMMARY parameter, detailed information about the specified triggers is displayed (Figure 10-4 on page 10-17, Table 10-2 on page 10-18).

The STATUS parameter displays general configuration information for the trigger facility (Figure 10-5 on page 10-19, Table 10-3 on page 10-19). A trigger identifier may not be specified.

The COUNTER parameter displays counters for the trigger facility (Figure 10-6 on page 10-20, Table 10-4 on page 10-20). A trigger identifier may not be specified.

Figure 10-3: Example output from the SHOW TRIGGER command.

TR#	Type & Details	Name	En	Te	Rept	#Scr	Days/Date
001	Periodic (3 min)	Test Trigger	Y	Y	Yes	01	-TW-FSS
002	Time (10:00)	Call home	Y	N	Yes	01	23-Apr-2000
003	Reboot (Crash)	Get Debug Info	Y	N	Yes	01	MTWTFSS

Table 10-1: Parameters displayed in the output of the SHOW TRIGGER command.

Parameter	Meaning
TR#	The trigger identifier (ID).
Type & Details	The trigger type and details; one of "Time" (trigger time), "Periodic" (period), "Reboot" (one of "Crash", "Reboot" or "All"), "Memory" or "CPU".
Name	The descriptive name of the trigger.
En	Whether or not the trigger is enabled; one of "Y" (Yes) or "N" (No).
Te	Whether or not the trigger is in test mode; one of "Y" (Yes) or "N" (No).
Rept	Whether or not the trigger repeats; one of "Y" (Yes), "N" (No), or a repeat count. The repeat count is decremented each time the trigger activates automatically.
#Scr	The number of scripts associated with the trigger.
Days/Date	The days or date on which the trigger will activate. For the days options the days are shown as a seven character string representing Monday to Sunday. Days on which the trigger will not activate are shown with a hyphen ("-").

Figure 10-4: Example output from the SHOW TRIGGER FULL command.

Trigger Configuration Details	

Trigger	1
Name	Bring up Wellington ISDN link
Type and details	Time (13:45)
Days	All
Enabled	Yes
Test	No
Repeat	No
Created/Modified	08-Nov-1996 12:04:33
Number of Activations	1
Last Activation	08-Nov-1996 13:45:07
Number of scripts	2
callwgtn.scp	
idlewgtn.scp	

Table 10-2: Parameters displayed in the output of the SHOW TRIGGER FULL command.

Parameter	Meaning
Trigger	The trigger identifier (ID).
Name	The descriptive name of the trigger.
Type and details	The trigger type and details; one of "Time" (trigger time), "Periodic" (period), "Reboot" (one of "Crash", "Reboot" or "All"), "Memory" or "CPU".
Days	A list of the days on which the trigger will activate, or one of "Weekdays" (Monday to Friday), "Weekends" (Saturday and Sunday) or "Daily" (every day). Only one of "Days" or "Date" will be displayed.
Date	The date on which the trigger will activate. Only one of "Days" or "Date" will be displayed.
Enabled	Whether or not the trigger is enabled; one of "Enabled" or "Disabled".
Test	Whether or not the trigger is in test mode; one of "Yes" or "No".
Repeat	Whether or not the trigger repeats; one of "Yes", "No" or a repeat count.
Created/Modified	The date and time the trigger was created or last modified.
Number of Activations	The number of times the trigger has been activated (triggered) since the last router restart.
Last Activation	The date and time the trigger was last activated (triggered).
Number of scripts	The number of scripts assigned to the trigger, followed by a list of the script file names.

Figure 10-5: Example output from the SHOW TRIGGER STATUS command.

```

Trigger Module Configuration
-----

General
Trigger Module ..... Enabled
Triggers configured ..... 4
Queued Commands ..... 0

Time Triggers
Configured ..... 2
Active ..... 2
Activated today ..... 1

Periodic Triggers
Configured ..... 1
Active ..... 1
Activated today ..... 0

Reboot Triggers
Configured ..... 0

Interface Triggers
Configured ..... 0

Resource Triggers
Configured ..... 1
Active ..... 1
Activated today ..... 0

```

Table 10-3: Parameters displayed in the output of the SHOW TRIGGER STATUS command.

Parameter	Meaning
General	General information about the Trigger Facility.
Trigger Module	Whether or not the trigger module is enabled; one of "Enabled" or "Disabled".
Triggers configured	The total number of triggers that have been configured.
Queued commands	The number of commands that are queued for execution.
Time Triggers	Information about time triggers.
Periodic Triggers	Information about periodic triggers.
Reboot Triggers	Information about reboot triggers.
Interface Triggers	Information about interface triggers.
Resource Triggers	Information about CPU and memory resource triggers.
Configured	The number of triggers of the associated type that have been configured.
Active	The number of triggers of the associated type that are currently active (enabled).
Activated today	The number of times a trigger of the associated type has been activated (triggered) today.

Figure 10-6: Example output from the SHOW TRIGGER COUNTER command.

```

Trigger Module Counters
-----
Polls (05 sec timer) ..... 37
Idle loop entry count ..... 5
Time trigger checks ..... 2
Time trigger queue rebuilds ..... 1
Trigger activations ..... 1
Time triggers activated today ..... 1
Periodic triggers activated today .. 0
Interface triggers activated today . 0
Resource triggers activated today .. 0

```

Table 10-4: Parameters displayed in the output of the SHOW TRIGGER COUNTER command.

Parameter	Meaning
Polls (05 sec timer)	The number of times the trigger module has polled for a trigger activation event.
Idle loop entry count	The number of times the trigger module has prepared commands for execution.
Time trigger checks	The number of times the trigger module has checked the list of time triggers for a trigger to activate.
Time trigger queue rebuilds	The number of times the time trigger queue has been rebuilt because time triggers have been added, deleted or modified, or because the time/date has been changed.
Trigger activations	The number of times a trigger has been activated.
Time triggers activated today	The number of times a time trigger has been activated today.
Periodic triggers activated today	The number of times a periodic trigger has been activated today.
Interface triggers activated today	The number of times an interface trigger has been activated today.
Resource triggers activated today	The number of times a CPU or memory resource trigger has been activated today.

Examples To display summary information for trigger 3, use the command:

```
SHOW TRIGGER=3 SUMMARY
```

To display summary information for all triggers, use the command:

```
SHOW TRIGGER
```

To display a detailed description of trigger 3, use the command:

```
SHOW TRIGGER=3
```

To display a detailed description of all triggers, use the command:

```
SHOW TRIGGER FULL
```

To display general configuration information for the trigger facility, use the command:

```
SHOW TRIGGER STATUS
```

To display counters for the trigger facility, use the command:

```
SHOW TRIGGER COUNTER
```

See Also ACTIVATE TRIGGER
 ADD TRIGGER
 CREATE TRIGGER
 DELETE TRIGGER
 DESTROY TRIGGER
 DISABLE TRIGGER
 ENABLE TRIGGER
 PURGE TRIGGER
 SET TRIGGER

Chapter 11

Time Division Multiplexing (TDM)

Introduction	11-2
Configuration Examples	11-2
Command Reference	11-3
ADD TDM	11-3
CREATE TDM	11-4
DELETE TDM	11-5
DESTROY TDM	11-5
PURGE TDM	11-6
SHOW TDM	11-6

Introduction

This chapter describes the Time Division Multiplexing (TDM) support provided by the router, and how to configure the router to use BRI TDM. Time division multiplexing is a mechanism for dividing the bandwidth of a link into separate channels or time slots. The router supports BRI TDM.

BRI TDM support is provided by a Basic Rate Interface. The interface can be used for Basic Rate ISDN and as a data channel for one or more static PPP links, with the restriction that the Basic Rate Interface has only 2 time slots of 64kbps. This chapter describes how to set up static TDM links over a Basic Rate Interface. See *Chapter 4, Integrated Services Digital Network (ISDN)* for detailed information about configuring ISDN Basic Rate interfaces.

A powerful feature of the router's support for TDM is the ability to use an interface for ISDN and static PPP links simultaneously. The slots available on the interface are statically apportioned, by command, between static TDM and dynamic ISDN use. For example slot 1 could be reserved for ISDN calls and slot 2 for a static TDM link. Note that not every telecommunication service provider is able to support simultaneous static and dynamic use of Basic Rate ISDN services.

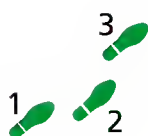
A Basic Rate interface provides 2 x 64kbps links (called B1 and B2) and one 16kbps link (called the D channel). The D channel is used for call control in ISDN applications. Some service providers use the Basic Rate technology to provide static links in which case the D channel is not used at all. Other services (German Monopol for example) provide standard ISDN on one 64kbps link and the D channel as well as one static 64kbps link.

Configuration Examples

The following example illustrates how to configure a Basic Rate interface for ISDN on the B1 and D channels and a static E1/T1 link on the B2 channel. Once the interface has been configured ISDN calls may be made over the interface in the same way as over a Basic Rate interface that is dedicated to ISDN operation. See *Chapter 4, Integrated Services Digital Network (ISDN)* for more information about ISDN calls. A TDM group is set up to use the B2 channel and a PPP interface is created to use the TDM group as its physical interface. The two steps of this process that are described here are:

- Configure a BRI port for ISDN and BRI TDM
- Create a TDM group

Once the TDM group has been created a PPP interface can be created to use the TDM group exactly as described in the previous example.



To configure a BRI port for ISDN and TDM:

1. Configure the physical parameters of the BRI port.

When a BRI link is used to provide a static link the service provider may disable normal activation and deactivation procedures since the presence of a static link implies that the interface should always be activated. This is accommodated with the command:

```
SET BRI=instance ACTIVATION=ALWAYS
```

where *instance* is the number of the BRI port.

2. Set the mode of the BRI interface.

The mode of the BRI port must be changed from the default of ISDN to MIXED, using the command:

```
SET BRI=instance MODE=MIXED ISDNSLOTS=1 TDMSLOTS=2
```

where *instance* is the number of the BRI port. The ISDNSLOTS parameter identifies the slot (B1) that is available for an ISDN call and the TDMSLOTS parameter identifies slot B2 as available for a TDM group.

3. Create a TDM group and assign a name.

The TDM group in this case will be called group2 and will use the B2 channel of the BRI0 interface. The group is created with the command:

```
CREATE TDM GROUP=group2 INTERFACE=bri0 SLOTS=2
```

Command Reference

This section describes the commands available on the router to configure and manage the TDM facility.

TDM requires the BRI module to be enabled and configured correctly. See *Chapter 4, Integrated Services Digital Network (ISDN)* for a detailed description of the commands required to enable and configure BRI interfaces.

See “Conventions” on page xxxv of *Preface* at the front of this manual for details of the conventions used to describe command syntax. See *Appendix A, Messages* for a complete list of messages and their meanings.

ADD TDM

Syntax `ADD TDM GROUP=groupname SLOTS=slotlist`

where:

- *groupname* is a character string, 1 to 15 characters in length. It may contain any alphanumeric character.
- *slotlist* is a character string defining a list of slots. It may include commas to separate individual time slots and dashes to indicate an inclusive range. For BRI interfaces valid slot numbers are 1 and 2, corresponding to the B1 and B2 channels respectively.

Description This command adds the specified time slots to the specified TDM group.

The GROUP parameter specifies the name of the TDM group to which the time slots are to be added. The group must have been created previously with the CREATE TDM command on page 11-4.

The SLOTS parameter specifies a comma-separated list of time slots to be added to the TDM group. A range of consecutive slots can be indicated by separating the first and last slots by a hyphen. The time slots may not already be in use by another TDM group. Slots can not be added to an unstructured TDM group.

Examples To add slot 2 to TDM group video, use the command:

```
ADD TDM GROUP=video SLOTS=2
```

See Also CREATE TDM
DELETE TDM
DESTROY TDM
PURGE TDM
SHOW TDM

CREATE TDM

Syntax CREATE TDM GROUP=*groupname* INTERFACE=*interface*
SLOTS=*slotlist*

where:

- *groupname* is a character string, 1 to 15 characters in length. It may contain any alphanumeric character.
- *interface* is the name of a BRI physical interface (e.g. BRI0).
- *slotlist* is a character string defining a list of slots. It may include commas to separate individual time slots and dashes to indicate an inclusive range. For BRI interfaces, valid slot numbers are 1 and 2, corresponding to the B1 and B2 channels respectively.

Description This command creates a new TDM group for the specified BRI interface, and associates one or more time slots of the interface with the group.

The GROUP parameter specifies the name of the TDM group to create. The name must be globally unique, that is, no other TDM group for any interface on the router may have the same name. More than one TDM group may be associated with the same interface, provided the TDM groups have different names and different lists of time slots.

The INTERFACE parameter specifies the interface with which the group is associated. The interface must be set to TDM or MIXED mode, using the SET BRI command on page 4-73 of *Chapter 4, Integrated Services Digital Network (ISDN)*.

The SLOTS parameter specifies a comma-separated list of time slots to be assigned to the TDM group. A range of consecutive slots can be indicated by separating the first and last slots by a hyphen. The time slots may not already be in use by another TDM group. The SLOTS and UNSTRUCTURED parameters are mutually exclusive and may not be specified together in the same command.

Examples To create a TDM group named “video”, with slot 2 of BRI port 1, use the command:

```
CREATE TDM GROUP=video INTERFACE=BRI1 SLOTS=2
```

See Also ADD TDM
DELETE TDM
DESTROY TDM
PURGE TDM
SHOW TDM

DELETE TDM

Syntax DELETE TDM GROUP=*groupname* SLOTS=*slotlist*

where:

- *groupname* is a character string, 1 to 15 characters in length. It may contain any alphanumeric character.
- *slotlist* is a character string defining a list of slots. It may include commas to separate individual time slots and dashes to indicate an inclusive range. For BRI interfaces valid slot numbers are 1 and 2, corresponding to the B1 and B2 channels respectively.

Description This command deletes one or more time slots from the specified TDM group.

The GROUP parameter specifies the name of the TDM group from which the time slots are to be deleted. The group must already exist.

The SLOTS parameter specifies a comma-separated list of the time slots to be deleted from the TDM group. A range of consecutive slots can be indicated by separating the first and last slots by a hyphen. The time slots must already be assigned to the specified TDM group and must not be in use by another TDM group. Slots can not be deleted from an unstructured TDM group.

Examples To delete slot 1 from the TDM group video, use the command:

```
DESTROY TDM GROUP=video SLOTS=1
```

See Also ADD TDM
CREATE TDM
DESTROY TDM
PURGE TDM
SHOW TDM

DESTROY TDM

Syntax DESTROY TDM GROUP=*groupname*

where:

- *groupname* is a character string, 1 to 15 characters in length. It may contain any alphanumeric character.

Description This command destroys the specified TDM group and releases all the time slots used by the group.

The GROUP parameter specifies the name of the TDM group to delete. The group must already exist, and no user modules may be attached to the group.

Examples To destroy the TDM group video, use the command:

```
DESTROY TDM GROUP=video
```

See Also ADD TDM
CREATE TDM
DELETE TDM
PURGE TDM
SHOW TDM

PURGE TDM

Syntax PURGE TDM GROUP

Description This command destroys all TDM groups. If any TDM group has a user module attached to it, the command will fail.

Examples To remove all TDM groups, use the command:

```
PURGE TDM GROUP
```

See Also ADD TDM
CREATE TDM
DELETE TDM
DESTROY TDM
SHOW TDM

SHOW TDM

Syntax SHOW TDM GROUP [=groupname] [INTERFACE=interface]

where:

- *groupname* is a character string, 1 to 15 characters in length. It may contain any alphanumeric character.
- *interface* is the name of a BRI physical interface (e.g. BRI0).

Description This command displays information about the TDM groups configured on the router (Figure 11-1 on page 11-7, Table 11-1 on page 11-7). Only groups associated with BRI ports set to TDM or MIXED mode will be displayed. If a TDM group name is specified then only information about that group will be displayed. The specified group must exist.

If the INTERFACE parameter is specified then only TDM groups associated with the specified interface will be displayed. The specified interface must exist and be set to TDM mode.

If both the group name and interface are specified, then the TDM group must be defined for the interface and the interface must be in TDM or MIXED mode.

Figure 11-1: Example output from the SHOW TDM GROUP command.

Interface	Group Name	User	Speed	Slots
bri0	group1	Yes	128K	1-2

Table 11-1: Parameters displayed in the output of the SHOW TDM GROUP command.

Parameter	Meaning
Group name	The name of the TDM group.
User	Whether or not a user module is attached to the TDM group; one of "Yes" or "No".
Speed	The aggregate speed of the group of slots.
Slots	The slots assigned to the TDM group.

Examples To display all the TDM groups that are defined on the router, use the command:

```
SHOW TDM GROUP
```

To display the TDM groups that are defined only for BRI interface 0, use the command:

```
SHOW TDM GROUP INTERFACE=BRI0
```

To display only the TDM group video, use the command:

```
SHOW TDM GROUP=video
```

See Also ADD TDM
CREATE TDM
DELETE TDM
DESTROY TDM
PURGE TDM

Chapter 12

Logging Facility

Introduction	12-2
Format of Log Messages	12-3
Secure Router Log Protocol	12-4
Net Manage Message Protocol	12-5
Processing of Log Messages	12-5
Output Definitions and Message Filters	12-5
Destinations	12-6
Configuring Output Definitions	12-8
Configuring Message Filters	12-8
Configuration Example	12-9
Command Reference	12-12
ADD LOG OUTPUT	12-12
ADD LOG RECEIVE	12-15
CREATE LOG OUTPUT	12-16
DELETE LOG OUTPUT	12-19
DELETE LOG RECEIVE	12-20
DESTROY LOG OUTPUT	12-20
DISABLE LOG	12-21
DISABLE LOG GENERATION	12-21
DISABLE LOG OUTPUT	12-21
DISABLE LOG RECEPTION	12-22
ENABLE LOG	12-22
ENABLE LOG GENERATION	12-22
ENABLE LOG OUTPUT	12-23
ENABLE LOG RECEPTION	12-23
FLUSH LOG OUTPUT	12-24
PURGE LOG	12-24
SET LOG OUTPUT	12-25
SET LOG RECEIVE	12-29
SET LOG UTCOFFSET	12-30
SHOW LOG	12-31
SHOW LOG COUNTERS	12-37
SHOW LOG OUTPUT	12-39
SHOW LOG QUEUE	12-43
SHOW LOG RECEIVE	12-44
SHOW LOG STATUS	12-45

Introduction

This chapter describes the logging facility provided by the router. The logging facility handles the generation, processing and display of log messages from the router. User-defined output definitions provide a powerful and flexible mechanism for filtering and prioritising log messages, and outputting selected messages to RAM, an asynchronous port on the router, or a UNIX syslog server. A secure router-to-router log message protocol (SRLP) enables log messages from regional and remote office routers to be forwarded to a central router for monitoring and processing.

The new logging facility is backwardly compatible with the old "Net Manage" log system. Net Manage log messages generated locally or forwarded via UDP port 5024 will be intercepted by the new logging facility and converted into the new log message format.

A router is a complex piece of computer equipment, combining both hardware and software. Multiple software modules operate simultaneously, with complex interactions between modules, processing large amounts of variable network traffic. It is often difficult to determine exactly what is happening "inside" a router that appears not to be operating correctly, or what was happening when problems occur.

A major task of network management is to monitor the operation of both permanent and on-demand network links (such as PPP links, ISDN calls and X.25 circuits), to maintain a high level of availability of network services, to record network usage and loading information for planning future developments, and for billing purposes.

The logging facility provides the network manager with a powerful, flexible and easily configurable tool for monitoring network activity and selecting and displaying the results.

The logging facility provides the following functions:

- Processing of log messages generated by any router module.
- Forwarding of log messages to other routers, and reception of log messages from other routers, via the Secure Router Log Protocol (SRLP, UDP port 5023).
- Reception of older Net Manage (UDP port 5024) log messages from other routers, or UNIX syslog messages, and conversion to the new log message format.
- Forwarding of selected log messages to a UNIX syslog server (UDP port 514).
- User-definable filters for selecting log messages.
- Storage of selected log messages in RAM.
- Output of selected log messages to an asynchronous port, in either full or summary format.
- Display of selected log messages stored in RAM, or messages queued for processing.

In particular the logging facility may be used to:

- Log critical router problems (`SEVERITY=>5`).
- Log interface status changes (`TYPE=VINT`).
- Log user login/authentication (`TYPE=AUTH`, `TYPE=USER`).
- Log trigger activity and script output (`TYPE=BATCH`).
- Log router commands (`TYPE=CMD`).
- Log router messages (`TYPE=MSG`).
- Log matches to IP filters, including IP header information and the contents of the data portion of IP packets (`TYPE=IPFILT`).

Format of Log Messages

A log message is a single entry in the router log, and is the fundamental unit of information processed by the logging facility. Each log message contains a number of data fields (Table 12-1 on page 12-3). A log message may contain accounting, user, debugging or other information as determined by the values of the log message fields. Depending on the type of log message generated, not all fields will contain a value.

Table 12-1: Log message fields.

Field	Size (bytes)	Description
Msg ID	4	The unique ID number for this message.
Flags	2	Flags and the message severity.
Date	2	The local date at which the message was generated (for the router which generated the message).
Time	3	The local time at which the message was generated (for the router which generated the message).
Origin IP	4	The IP address of the originator of the message.
Module	2	The ID of the module generating the message.
Type	2	The type of the message.
SubType	2	The subtype of the message.
Source File	12	The file name of the source file where the message originated.
Source Line	2	The line number in the source file where the message originated.
Reference	15	The Reference ID (e.g. user name, ISDN call name).
Message	80	The message text.

The *Flags* field contains control flags and the *Severity* of the log message. Severity is expressed as a number in the range 0 to 7 (Table 12-2 on page 12-4).

Table 12-2: Log message severity levels.

Severity	Value	Description
CRITICAL	7	Router operation severely impaired.
URGENT	6	Router operation has been or could be affected.
IMPORTANT	5	Issue that requires manager attention, possible problem.
NOTICE	4	Issue that may require manager attention.
INFO	3	Normal notification of an event, not serious or particularly important.
DETAIL	2	Useful information, can be ignored during normal operation.
TRIVIAL	1	Generally unimportant everyday events.
DEBUG	0	Extremely detailed (possibly high-volume) debugging information.

The *Date* and *Time* fields contain the date and time that the log message was generated. All log messages are stored and processed using UTC (Universal Coordinated Time) so that routers in different time zones may sensibly share log messages. As of Software Release 7.4 the router's software is Year 2000 compliant.

The *Module* field contains the module identifier of the module which generated the log message. See “*Module Identifiers and Names*” on page B-2 of *Appendix B, Reference Tables* for a complete list.

The *Type* and *SubType* fields contain the type and subtype identifiers for the log message. See “*Log Message Types and Subtypes*” on page B-8 of *Appendix B, Reference Tables* for a complete list. The *Type* field identifies the general category of event which triggered the generation of the log message, and the *SubType* field identifies a specific event within that category. Types and subtypes may be specified or displayed either by numeric identifier or by name. See “*Log Message Types and Subtypes*” on page B-8 of *Appendix B, Reference Tables* for a complete list of IDs and names. The values of the *Type* and *SubType* fields also determine the contents of the *Reference* and *Message* fields.

Secure Router Log Protocol

The logging facility provides an extensible log message protocol, the *Secure Router Logging Protocol (SRLP)*, to permit the secure exchange of log messages between routers.

A log message is encoded into a UDP datagram with a checksum and an MD5 authentication digest, and transmitted to UDP port 5023. Since UDP is an unreliable transport medium, each log message must be acknowledged by the receiver. The acknowledgements (ACKs) are also UDP datagrams transmitted to UDP port 5023. Unacknowledged messages are retransmitted after 1, 4, 16, 64 and 256 minutes.

The UDP packets are protected by encryption, preventing them being read by unauthorised parties, and can be authenticated using passwords and MD5 digests.

Net Manage Message Protocol

The logging facility will accept log messages from routers using the old “Net Manage” UDP logging protocol on port 5024. The logging facility will **not** send log messages back to routers using this protocol, but it will generate the ACKs required to acknowledge reception of the Net Manage messages. No other Net Manage facilities are supported.

Net Manage messages received via the Net Manage logging protocol or generated locally are converted to the new log message format. Log message fields with no equivalent in the old Net Manage message format are set to default values. In particular, the *Type* and *Subtype* fields are set to NULL (displayed as a blank in output) and the *Severity* field is set to 0.

Processing of Log Messages

The processing of log messages is controlled by user-defined log message filters and output definitions.

Output Definitions and Message Filters

A *log message filter* is a set of conditions on the fields of a log message. Log messages meeting these conditions are said to “match” the filter. Log message filters are used to specify which log entries should be displayed (using the SHOW LOG command on page 12-31), accepted for further processing by an output definition, or ignored.

An *output definition* describes the processing to be performed on log messages that match one of the log message filters associated with the output definition. Log messages can be stored in RAM, output to an asynchronous port on the router, forwarded to another router via the Secure Router Logging Protocol (SRLP), or forwarded to a UNIX syslog server. An output definition may have one or more associated log message filters.

The logging facility can receive Net Manage messages locally generated or via the Net Manage protocol on UDP port 5024, syslog messages on UDP port 514, and new-format log messages generated locally or transmitted via the new Secure Router Logging Protocol (SRLP) on UDP port 5023. Net Manage and syslog messages are automatically converted to the new log message format. When a log message is received, the logging facility checks to see whether the log message can be processed by one or more of the output definitions.

For each output definition defined, the log message filters associated with the output definition are applied in sequence. If the log message matches a filter with an IGNORE action, processing continues with the next output definition (if any). If the log message does not match any filter, processing continues with the next output definition (if any). If the log message matches a filter with a PROCESS action, the log message is processed according to the output definition and then processing continues with the next output definition (if any).

A single log message may be processed more than once. For example, all regional and remote office routers could be configured to forward all log messages to a central site router. At the central site router, all log messages

could be stored in RAM and output to an asynchronous port to which a printer is attached. In addition, all low severity log messages relating to user activity (e.g. logins) and on-demand links (e.g. normal call establishment and clearing) could be forward to a syslog server on an accounting host, while all high severity log messages are forwarded to a syslog server on a network management station.

Each output definition has its own separate message queue. When a log message matches a filter associated with an output definition, a copy of the log message is placed on the output definition queue as part of the processing performed by the output definition. The function of the queue varies depending on the output definition. For output definitions that store log messages in RAM, the queue represents the actual log messages stored in RAM. The SHOW LOG command on page 12-31 simply displays the contents of the queue. For output definitions that forward log messages to an asynchronous port, to another router via SRLP, or to a syslog server, the queue represents the log messages waiting to be processed (or acknowledged).

The flexibility of the output definition mechanism means that it is possible to create two or more output definitions with the same destination (e.g. a syslog server) but with different filters. As a result, the logging facility will maintain two or more separate queues of log messages, all of which are waiting to be forwarded to the same syslog server. This flexibility could potentially cause problems for output definitions that store messages in RAM. If multiple queues are stored in RAM, which one is displayed by the SHOW LOG command? The conflict is solved by the provision of a special output definition, TEMPORARY. The TEMPORARY output definition only accepts a destination of MEMORY (RAM). Messages processed by this output definition can be displayed by the SHOW LOG and SHOW LOG=TEMPORARY commands.

Destinations

Storage in RAM

Log messages may be stored in the router's RAM memory. Log messages stored in RAM are **not** retained over a power failure or router restart. There is no preset limit on the number of log messages that can be stored in RAM, except that log messages will not be stored in RAM when the number of buffers falls below Buffer Level 2. However, the maximum number of messages that may be stored in RAM at any one time can be configured. When the number of messages reaches the maximum, the oldest message is deleted to make room for a new message.

Output to an Asynchronous Port

Log messages may be output to an asynchronous port, which may be connected to a serial printer, terminal or other serial device. The log messages may be displayed in either summary or full format.

Forwarding to Another Router Via SRLP

Log messages may be transferred to a central router for display, processing and output, using the Secure Router Logging Protocol (SRLP). The messages transferred to the remote router will appear intact, with no information loss. The remote router will know where the log message came from, and this can be displayed in the SHOW LOG FULL output.

Forwarding to a Unix Syslog Server

Log messages may be converted to Unix syslog format and transmitted to a UNIX-style logging daemon, normally called syslogd, on a host accessible via IP. Syslog is a system logging facility provided by many versions of UNIX. Some translation is performed to match the database-like structure of the router's log message format to the textual format of syslog records.

The type and subtype codes are translated into syslog facility identifiers (Table 12-3 on page 12-7) and the log message severity is translated into a syslog "level" (Table 12-4 on page 12-7). When converted to syslog textual format, the module ID in the new log message format is converted to a short module abbreviation at the start of the syslog message.

The syslog-format messages are transmitted via UDP to the syslog port of the defined syslog server. The syslog protocol does not support message encryption, authentication or reliable delivery (acknowledgements).

Table 12-3: Mapping between logging facility module identifier, type and subtype, and syslog facility identifiers.

Type	Facility	Meaning
000/NULL	LOG_USER	Log messages without a type (old message format)
010/LIC		Licencing information.
011/AUTH	LOG_AUTH	Authentication and security issues.
012/TRIG	LOG_CRON	Time-based activities (triggers and output).
013/LPR	LOG_LPR	Line Printer Daemon activity.
001/REST	LOG_LOCAL7	Router restarts.
008/EXCEP		Exceptions.
009/BUFF		Buffer issues.
002/PINT	LOG_LOCAL6	Physical interface and data-link issues.
003/DLINK		
004/CALL	LOG_LOCAL5	ISDN call issues.
005/VINT		Virtual Interface issues.
006/CIRC	LOG_LOCAL4	Circuit and PPP CP issues.
007/ATT		Attachments.

Table 12-4: Mapping between logging facility severity levels and syslog levels.

Severity	Syslog Level	Meaning
7	LOG_ALERT	Router error, function impaired.
6	LOG_CRIT	Critical link or interface problem, may not work.
5	LOG_ERR	Not-so-serious error or authentication issue.
4	LOG_ERR	Less serious problem or authentication warning.
3	LOG_WARNING	Possible problem with interface or configuration.
2	LOG_NOTICE	Fairly important informational message/tracing.
1	LOG_INFO	Less important informational message/tracing.
0	LOG_DEBUG	Trivial debugging or tracing.

Configuring Output Definitions

By default, logging is enabled, and the special output definition is defined. The TEMPORARY output definition contains a log message filter that matches all log messages of severity 3 or greater, and stores up to 300 messages in RAM.

An output definition is created using the command:

```
CREATE LOG OUTPUT={TEMPORARY|output-id} DESTINATION={MEMORY|
PORT|ROUTER|SYSLOG} [FORMAT={FULL|MSGONLY|SUMMARY}]
[MAXQUEUESEVERITY=severity] [MESSAGES=message-count]
[PASSWORD={password|NONE}] [PORT=port-number]
[QUEUEONLY={YES|NO}] [SECURE={YES|NO}] [SERVER=ipadd]
[ZONE={time-zone-name|utc-offset}]
```

A log message filter must be defined for the output definition before the output definition can process any log messages. By default, output definitions are enabled when they are created. Output definitions can be temporarily disabled or enabled using the commands:

```
DISABLE LOG OUTPUT={TEMPORARY|output-id}
ENABLE LOG OUTPUT={TEMPORARY|output-id}
```

An existing output definition can be modified using the command:

```
SET LOG OUTPUT={TEMPORARY|output-id} [DESTINATION={MEMORY|
PORT|ROUTER|SYSLOG}] [FORMAT={FULL|MSGONLY|SUMMARY}]
[MAXQUEUESEVERITY=severity] [MESSAGES=message-count]
[PASSWORD={password|NONE}] [PORT=port-number]
[QUEUEONLY={YES|NO}] [SECURE={YES|NO}] [SERVER=ipadd]
[ZONE={time-zone-name|utc-offset}]
```

An output definition can be deleted (destroyed) using the command:

```
DESTROY LOG OUTPUT={TEMPORARY|output-id}
```

The currently configured output definitions can be displayed using the command:

```
SHOW LOG OUTPUT[={TEMPORARY|output-id}] [{FILTER=filter-id|
FULL}]
```

Configuring Message Filters

When an output definition is created, it has no associated log message filters and therefore no log messages will be selected for processing by the output definition. At least one log message filter must be defined and associated with the output definition before the output definition will become active (start processing messages).

An log message filter is created and associated with an output definition using the command:

```
ADD LOG OUTPUT={TEMPORARY|output-id} [FILTER=filter-id]
[ACTION={PROCESS|IGNORE}] [ALL] [DATE=[op] dd-mm-yyyy]
[DEVICE=[op] device] [FILE=[op] filename] [MASK=ipadd]
[MSGTEXT=[op] string] [MODULE=[op] module-id] [ORIGIN=ipadd]
[REFERENCE=[op] string] [SEVERITY=[op] severity]
[SOURCELINE=[op] line] [SUBTYPE=[op] subtype-id]
[TIME=[op] hh:mm:ss] [TYPE=[op] type-id]
```

An existing log message filter can be modified using the command:

```
SET LOG OUTPUT={TEMPORARY|output-id} FILTER=filter-id
[ACTION={PROCESS|IGNORE}] [ALL] [DATE=[op] dd-mm-yyyy]
[DEVICE=[op] device] [FILE=[op] filename] [MASK=ipadd]
[MSGTEXT=[op] string] [MODULE=[op] module-id] [ORIGIN=ipadd]
[REFERENCE=[op] string] [SEVERITY=[op] severity]
[SOURCELINE=[op] line] [SUBTYPE=[op] subtype-id]
[TIME=[op] hh:mm:ss] [TYPE=[op] type-id]
```

Most parameters support additional operators (<, >, !=, %) between the equals sign (“=”) and the value that modify the comparison between the value in the filter and the value in the log message field (Table 12-5 on page 12-9).

Table 12-5: Log message filter comparison operators.

Operator	Example	Meaning
< Less than	SEVERITY=<5	The log message matches if the value in the log message field is less than the value specified in the filter.
> Greater than	DEVICE=>1	The log message matches if the value in the log message field is greater than the value specified in the filter.
! Not equal	TYPE=!2	The log message matches if the value in the log message field is not equal to the value specified in the filter.
(none) Equal	MOD=PPP	The log message matches if the value in the log message field is equal to the value specified in the filter.
% Contains substring	REF=%call	The log message matches if the value in the log message field contains the value specified in the filter (string fields only).

A log message filter can be deleted using the command:

```
DELETE LOG OUTPUT={TEMPORARY|ALL|output-id} FILTER={ALL|filter-id}
```

The currently configured log message filters definitions can be displayed using the command:

```
SHOW LOG OUTPUT[={TEMPORARY|output-id}] {FILTER=filter-id|FULL}
```

Configuration Example

The following example illustrates the steps required to configure the logging facility in a wide area network environment (Figure 12-1 on page 12-10, Table 12-6 on page 12-10). A router at the Remote Office is connected via a wide area link to a router at the Head Office. The remote router will be configured to forward all log messages to the Head Office router via the Secure Router Logging Protocol (SRLP) with password authentication. At the Head Office router, all log messages are to be stored in RAM. In addition, all log messages relating to ISDN calls will be forwarded to a syslog server for accounting purposes, and all critical log messages will be forwarded via an asynchronous port to a network management station.

Figure 12-1: Example configuration for a basic logging facility.

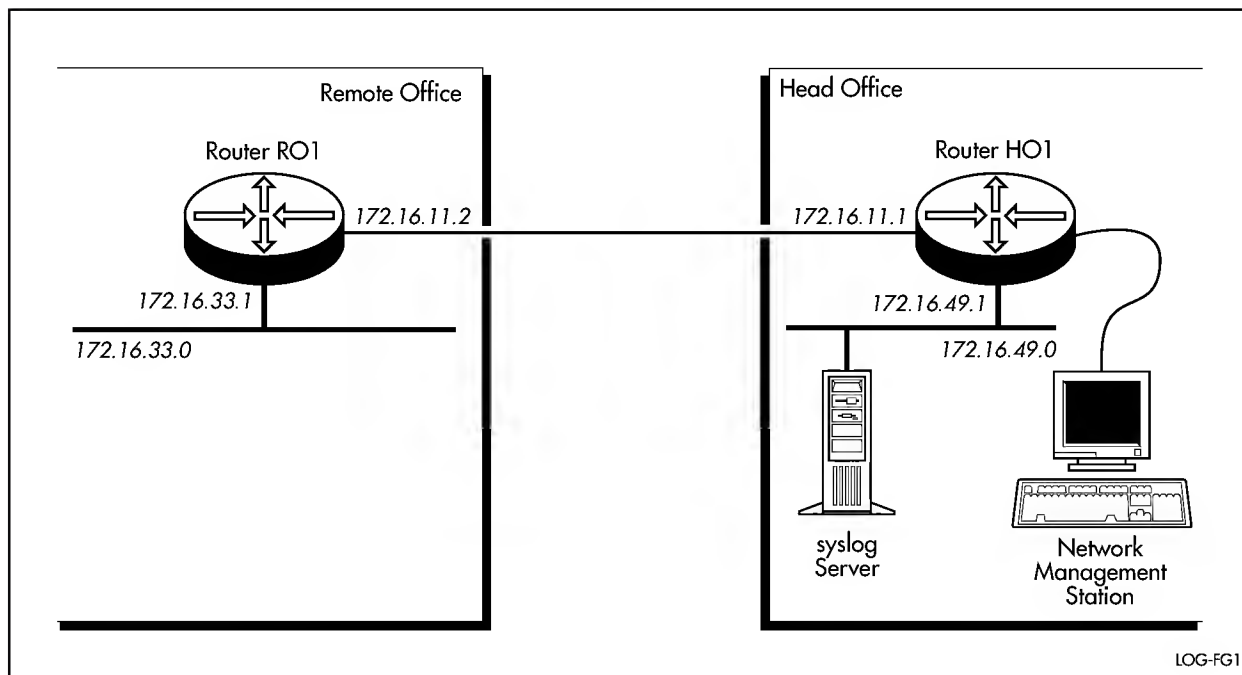
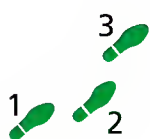


Table 12-6: Example configuration parameters for a basic logging facility.

Parameter	Head Office	Remote Office
Router name	HO1	RO1
IP address of LAN	172.16.49.0	172.16.33.0
IP address of Ethernet interface eth0	172.16.49.1	172.16.33.1
IP address of PPP link	172.16.11.0	172.16.11.0
IP address of PPP interface ppp0	172.16.11.1	172.16.11.2
NMS connected to asynchronous port	1	-
IP address of syslog server	172.16.49.8	-

**To configure the Remote Office router:****1. Enable the logging facility.**

By default, logging is enabled. Check that this is the case using the command:

```
SHOW LOG STATUS
```

If necessary enable logging and the generation of log messages:

```
ENABLE LOG
ENABLE LOG GENERATION
```

2. Create an output definition and a message filter.

Create an output definition to forward log messages via the Secure Router Logging Protocol (SRLP) to the Head Office router, with password authentication. The SECURE option defaults to YES when DESTINATION is set to ROUTER and a password is specified, so security-related messages (e.g. password changes) will be processed by this output definition:

```
ENABLE IP
```



```
CREATE LOG OUTPUT=1 DESTINATION=ROUTER SERVER=172.16.11.1
PASSWORD=GB4La8z
```

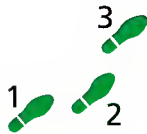
Add a log message filter that matches any log message to the output definition so that all log messages generated on the router will be processed by this output definition. The **FILTER** parameter is optional; by default the filter is added to the end of the list of filters:

```
ADD LOG OUTPUT=1 FILTER=1 ALL
```

3. Check the configuration.

Check that the output definition and message filter configuration is correct by displaying the output definition and its associated filters:

```
SHOW LOG OUTPUT=1 FULL
```



To configure the Head Office router:

1. Enable the logging facility.

By default, logging is enabled. Check that this is the case using the command:

```
SHOW LOG STATUS
```

If necessary enable logging and the generation of log messages:

```
ENABLE LOG
ENABLE LOG GENERATION
```

2. Enable the reception of log messages from the remote router.

Enable the reception of log messages via SRLP, and add the remote router to the log reception table so that log messages will be accepted from the remote router only with password authentication:

```
ENABLE LOG RECEPTION
ADD LOG RECEIVE=172.16.11.2 PASSWORD=GB4La8z PROT=NEW
```

3. Create output definitions and message filters.

The **TEMPORARY** output definition is automatically defined by the system to match all log messages of severity 3 or greater and to store the log messages in RAM. Since this is exactly what is required in this example, there is no need to specify this output definition any further.

Create an output definition and associated message filter to output all critical log messages to asynchronous port 1:

```
CREATE LOG OUTPUT=1 DESTINATION=PORT PORT=1 FORMAT=FULL
ADD LOG OUTPUT=1 FILTER=1 SEVERITY=>5
```

Create an output definition and associated message filter to output all call-related log messages to the syslog server:

```
CREATE LOG OUTPUT=2 DESTINATION=SYSLOG SERVER=172.16.49.8
MESSAGES=20
ADD LOG OUTPUT=2 FILTER=1 TYPE=CALL
```

4. Check the configuration.

Check that the output definition and message filter configuration is correct by displaying the output definitions and their associated filters:

```
SHOW LOG OUTPUT FULL
```

Command Reference

This section describes the commands to configure and manage the logging facility in the router.

Some features and options of the logging facility require the IP module to be enabled and configured correctly. See *Chapter 6, Internet Protocol (IP)* for detailed descriptions of the commands required to enable and configure IP.

See “Conventions” on page xxxv of *Preface* in the front of this manual for details of the conventions used to describe command syntax. See *Appendix A, Messages* for a complete list of messages and their meanings.

ADD LOG OUTPUT

Syntax `ADD LOG OUTPUT={TEMPORARY|output-id} [FILTER=filter-id]
 [ACTION={PROCESS|IGNORE}] [ALL] [DATE=[op] dd-mmm-yyyy]
 [DEVICE=[op] device] [FILE=[op] filename] [MASK=ipadd]
 [MSGTEXT=[op] string] [MODULE=[op] module-id]
 [ORIGIN=ipadd] [REFERENCE=[op] string]
 [SEVERITY=[op] severity] [SOURCELINE=[op] line]
 [SUBTYPE=[op] subtype-id] [TIME=[op] hh:mm:ss]
 [TYPE=[op] type-id]`

where:

- *output-id* is the index number of an output definition, in the range 1 to 20.
- *filter-id* is a filter entry number, in the range 1 to *n*+1, where *n* is the number of filters currently defined for the output definition.
- *op* is a comparison operator (see Table 12-5 on page 12-9).
- *dd-mmm-yyyy* is a date, where *dd* is the day number (1–31), *mmm* is a three-letter abbreviation for the month (“Jan”, “Feb”, “Mar”, ...) or the month number (1–12), and *yyyy* is the year.
- *device* is a router device number.
- *filename* is a module source file name, 1 to 12 characters in length.
- *ipadd* is an IP address in dotted decimal notation.
- *module-id* is the name or number of a router module (see “Module Identifiers and Names” on page B-2 of *Appendix B, Reference Tables* for a complete list).
- *string* is a character string, 1 to 15 characters in length.
- *severity* is a log message severity, in the range 0 (low) to 7 (high).
- *line* is a line number in a module source file, in the range 1 to 65535.
- *subtype-id* is the name or number of a log message subtype (see “Log Message Types and Subtypes” on page B-8 of *Appendix B, Reference Tables* for a complete list).
- *hh:mm:ss* is a time, where *hh* is the hour (0–23), *mm* is the minutes (0–59), and *ss* is the seconds (0–59).

- *type-id* is the name or number of a log message type (see “Log Message Types and Subtypes” on page B-8 of Appendix B, Reference Tables for a complete list).

Description This command adds a log filter to the specified output definition. The log filter specifies a set of conditions that must hold for a log entry to *match* the filter, and whether or not to *process* or *ignore* matching log messages. If there are no conditions specified, the filter matches nothing.

The OUTPUT parameter specifies the number of the output definition to which the filter entry is to be added. The output definition must already exist. If TEMPORARY is specified, the filter is added to the special TEMPORARY output definition.

The FILTER parameter specifies the entry number of the filter within the output definition. If FILTER is specified, the filter will be inserted into the filter list at the specified position. If FILTER is not specified, the filter will be added to the end of the filter list for the output definition.

The ACTION parameter specifies the action to perform for log messages matching this filter. If PROCESS is specified, the log message is processed according to the output definition. If IGNORE is specified, the log message is ignored and not processed by this output definition. The default is PROCESS.

The ALL parameter matches all log entries. If ALL is specified, no other selection criteria may be specified for this filter. The default is to only match log entries fitting the specified criteria.

The DATE parameter specifies the date value to match in the log message. The first character of the value may be one of the comparison operators “<”, “>” or “!” to modify the comparison from “equals” (the default) to “less than or equal to”, “greater than or equal to” or “not equal to” respectively (see Table 12-5 on page 12-9). The default is to match any date.

The DEVICE parameter specifies the device number to match in the log message. The first character of the value may be one of the comparison operators “<”, “>” or “!” to modify the comparison from “equals” (the default) to “less than or equal to”, “greater than or equal to” or “not equal to” respectively (see Table 12-5 on page 12-9). The default is to match any device number.

The FILE parameter specifies the name of a source file to match in the log message. The first character of the value may be one of the comparison operators “<”, “>” or “!” to modify the comparison from “equals” (the default) to “less than or equal to”, “greater than or equal to” or “not equal to” respectively (see Table 12-5 on page 12-9). The default is to match any source file.

The MASK parameter specifies a subnet mask to use in association with the ORIGIN IP address parameter. The default is 255.255.255.255.

The MSGTEXT parameter specifies a string to match in the text of the log message. The first character of the value may be one of the comparison operators “<”, “>”, “!” or “%” to modify the comparison from “equals” (the default) to “less than or equal to”, “greater than or equal to”, “not equal to” or “contains substring” respectively (see Table 12-5 on page 12-9). The default is to match any text.

The MODULE parameter specifies the router module to match in the log message, as either a decimal number or a recognised module name (see *"Module Identifiers and Names"* on page B-2 of *Appendix B, Reference Tables* for a complete list). The first character of the value may be one of the comparison operators "<", ">" or "!" to modify the comparison from "equals" (the default) to "less than or equal to", "greater than or equal to" or "not equal to" respectively (see Table 12-5 on page 12-9). The default is to match any module.

The ORIGIN parameter specifies the IP address to match against the originating IP address field of the log message. The MASK parameter can be used to specify a host, subnet or network. The default is to match any IP address in the origin IP address field.

The REFERENCE parameter specifies the reference to match in the log message. The first character of the value may be one of the comparison operators "<", ">", "!" or "%" to modify the comparison from "equals" (the default) to "less than or equal to", "greater than or equal to", "not equal to" or "contains substring" respectively (see Table 12-5 on page 12-9). The default is to match any reference.

The SEVERITY parameter specifies the log message severity level to match in the log message. The first character of the value may be one of the comparison operators "<", ">" or "!" to modify the comparison from "equals" (the default) to "less than or equal to", "greater than or equal to" or "not equal to" respectively (see Table 12-5 on page 12-9). The default is to match any severity.

The SOURCELINE parameter specifies the line number in the source file where the log message was generated, to match in the log message. The first character of the value may be one of the comparison operators "<", ">" or "!" to modify the comparison from "equals" (the default) to "less than or equal to", "greater than or equal to" or "not equal to" respectively (see Table 12-5 on page 12-9). The default is to match any source line number.

The SUBTYPE parameter specifies the log message subtype to match, as either a decimal number or a recognised subtype name (see *"Log Message Types and Subtypes"* on page B-8 of *Appendix B, Reference Tables* for a complete list). The first character of the value may be one of the comparison operators "<", ">" or "!" to modify the comparison from "equals" (the default) to "less than or equal to", "greater than or equal to" or "not equal to" respectively (see Table 12-5 on page 12-9). The default is to match any log message subtype.

The TIME parameter specifies the time value to match in the log message. The first character of the value may be one of the comparison operators "<", ">" or "!" to modify the comparison from "equals" (the default) to "less than or equal to", "greater than or equal to" or "not equal to" respectively (see Table 12-5 on page 12-9). The default is to match any time.

The TYPE parameter specifies the log message type to match, as either a decimal number or a recognised type name (see *"Log Message Types and Subtypes"* on page B-8 of *Appendix B, Reference Tables* for a complete list). The first character of the value may be one of the comparison operators "<", ">" or "!" to modify the comparison from "equals" (the default) to "less than or equal to", "greater than or equal to" or "not equal to" respectively (see Table 12-5 on page 12-9). The default is to match any log message type.



Any parameter values that contain spaces must be enclosed in double quotes. If one of the operators ("<", ">", "!", "%") is also present, the operator must be inside the quote marks. For example, `ADD LOG OUTPUT=4 MSGTEXT="%PPP Inter"`.

Examples To add a filter to output definition 17 that causes all log messages of severity less than 6 to be ignored, use the command:

```
ADD LOG OUTPUT=17 SEVERITY=<6 ACTION=IGNORE
```

See Also DELETE LOG OUTPUT
SET LOG OUTPUT
SHOW LOG OUTPUT

ADD LOG RECEIVE

Syntax `ADD LOG RECEIVE={ipadd|ANY} [ALLOW={YES|NO}] [MASK=ipadd]
[PASSWORD={password|NONE}] [PROTOCOL={ALL|BOTH|NEW|OLD|
SYSLOG}]`

where:

- *ipadd* is an IP address in dotted decimal notation.
- *password* is a character string, 1 to 16 characters in length. Valid characters are any printable character. If the string contains spaces it must be enclosed in double quotes.

Description This command adds an entry to the log reception table. The log reception table, when not empty, specifies the routers (and networks/subnets if MASK is specified) from which log messages will be accepted. If the log reception table is empty (the default), log messages will not be accepted from any source. When comparing the source addresses of received log messages, the most specific entry is used. The order of entries in the log reception table is not significant.

The RECEIVE parameter specifies the IP address of the host, subnet or network from which log messages will be received. If ANY is specified, a wildcard entry is added to accept log messages from any IP address. If more than one receive entry matches an IP address, the most specific entry (the one with the most specific network mask) is used.

The ALLOW parameter specifies whether or not log messages will be accepted from the specified IP address. If YES is specified, log messages will be accepted from the IP address. If NO is specified, log messages will not be accepted from the IP address. The default is YES.

The MASK parameter specifies a subnet mask to use in association with the RECEIVE parameter. The default is 255.255.255.255 if an IP address is specified for the RECEIVE parameter, or 0.0.0.0 if ALL is specified for the RECEIVE parameter.

The PASSWORD parameter specifies the password that must accompany log messages from the specified IP address, for authentication purposes when log messages are forwarded to another router via the Secure Router Logging Protocol (SRLP). If the PASSWORD option is present, the specified password must accompany log messages from the specified IP address.

The **PROTOCOL** parameter specifies the protocol to use for message reception from the specified IP address. If **OLD** is specified the logging facility will accept old Net Manage (UDP port 5024) packets. If **NEW** is specified the logging facility will accept the new Secure Router Logging Protocol (SRLP) packets. If **SYSLOG** is specified the logging facility will accept syslog messages. The **BOTH** option is equivalent to specifying both **OLD** and **NEW**. The **ALL** option is equivalent to specifying **OLD**, **NEW** and **SYSLOG**. The **PASSWORD** parameter is not valid when **TYPE** is set to **OLD** or **SYSLOG**, as password authentication is not supported in these protocols.

Examples To ensure that only log messages from network 192.168.0.0 are processed, use the command:

```
ADD LOG RECEIVE=192.168.0.0 MASK=255.255.0.0 PROTOCOL=BOTH
```

To accept messages from subnet 192.168.2.0 only with password **SECRET**, use the command:

```
ADD LOG RECEIVE=192.168.2.0 MASK=255.255.255.0
PASSWORD=SECRET PROTOCOL=NEW
```

See Also DELETE LOG RECEIVE
SET LOG RECEIVE
SHOW LOG RECEIVE

CREATE LOG OUTPUT

Syntax CREATE LOG OUTPUT={TEMPORARY|*output-id*}
DESTINATION={MEMORY|PORT|ROUTER|SYSLOG} [FORMAT={FULL|MSGONLY|SUMMARY}] [MAXQUEUESEVERITY=*severity*]
[MESSAGES=*message-count*] [PASSWORD={*password*|NONE}]
[PORT=*port-number*] [QUEUEONLY={YES|NO}] [SECURE={YES|NO}] [SERVER=*ipadd*] [ZONE={*time-zone-name*|*utc-offset*}]

where:

- *output-id* is the index number of an output definition, in the range 1 to 20.
- *severity* is a message severity level, in the range 0 (low) to 7 (high).
- *message-count* is the maximum number of log messages that may be queued for processing.
- *password* is a character string, 1 to 16 characters in length. Valid characters are any printable character. If the string contains spaces it must be enclosed in double quotes.
- *port-number* is the number of an asynchronous port. Ports are numbered sequentially starting with port 0.
- *ipadd* is an IP address in dotted decimal notation.
- *time-zone-name* is the name of a recognised time zone (Table 12-7 on page 12-18).
- *utc-offset* is a time offset from GMT/UTC, in the range +23:59:59 to 23:59:59.

Description This command creates an output definition, which specifies the processing to be performed on log messages that match one of the log message filters associated with the output definition. The specified output definition must not

already exist. Once the output definition has been created, log message filters are added using the ADD LOG OUTPUT command on page 12-12.

The OUTPUT parameter specifies the index number of the output definition to be created, or the special output definition TEMPORARY. If TEMPORARY is specified, the parameters MAXQUEUESEVERITY, QUEUEONLY and SECURE may not be specified. An output definition must not already exist with this index number.

The DESTINATION parameter specifies the type of processing to be performed and the destination of log messages processed by this output definition. The MEMORY option stores log messages in RAM. The MEMORY option is only valid only when OUTPUT is set to TEMPORARY. The PORT option outputs log messages to an asynchronous port on the router. If DESTINATION is set to PORT, the parameters MAXQUEUESEVERITY, MESSAGES and QUEUEONLY may not be specified. The ROUTER option forwards log messages via the Secure Router Logging Protocol (SRLP) to another router. The SYSLOG option forwards log messages in syslog format to a syslog server.

The FORMAT parameter specifies the format of log messages when converted into ASCII text for output to an asynchronous port. The FULL option displays the entire log message in multiple lines, with a blank line between messages. The SUMMARY option produces an abbreviated display. The MSGONLY option displays only the text of the message. The FORMAT parameter is valid only if DESTINATION is set to PORT. The default is to display a summary of each log message on a single line, omitting some fields.

The MAXQUEUESEVERITY parameter specifies the maximum message severity level at which messages will be queued but not output, when QUEUEONLY is set to YES. If QUEUEONLY is set to YES, log messages with a severity level less than that specified by the MAXQUEUESEVERITY parameter are queued until the queue length reaches the limit set by the MESSAGES parameter, at which point all the messages are processed. Any log messages with a priority greater than the value of MAXQUEUESEVERITY will cause the queued messages to be flushed (processed). If the DESTINATION parameter is set to PORT or the OUTPUT parameter is set to TEMPORARY, MAXQUEUESEVERITY may not be specified. The default for MAXQUEUESEVERITY is 7 (i.e. only messages with the maximum severity level are output immediately).

The MESSAGES parameter specifies the number of log messages that will be added to the output definition queue before actually being processed. For a DESTINATION of MEMORY, the MESSAGES parameter specifies the maximum number of messages to be stored. When the limit is reached, older messages are purged to make room for new messages. For a DESTINATION of SYSLOG or ROUTER, the MESSAGES parameter specifies the maximum number of messages awaiting processing or acknowledgement. The MESSAGE parameter is not permitted if DESTINATION is set to PORT. For SYSLOG and ROUTER, the MAXQUEUESEVERITY parameter can cause high-priority messages (and any other queued messages) to be output immediately. The default is 300 for a DESTINATION of RAM, and 20 for a DESTINATION of ROUTER or SYSLOG.

The PASSWORD parameter specifies the password to attach to log messages for authentication purposes when log messages are forwarded to another router via the Secure Router Logging Protocol (SRLP). If the remote router requires a password, this password must match. The PASSWORD parameter is only valid when DESTINATION is set to ROUTER. The password is not transmitted over the network, but is used to compute an MD5 digest. The default is no password.

The PORT parameter specifies an asynchronous port on the router to which log messages are to be directed. The PORT parameter is only valid, and is required, when DESTINATION is set to PORT.

The QUEUEONLY parameter controls the output of log messages from the output definition queue. When QUEUEONLY is set to YES log messages are queued by the output definition and are not actually processed (forwarded, printed, displayed, etc.) until the queue is full. If the DESTINATION parameter is set to PORT or the OUTPUT parameter is set to TEMPORARY, QUEUEONLY may not be specified. The default is NO.

The SECURE parameter specifies whether or not messages processed through this output definition will be “secure” (the meaning of the word “secure” in this context is defined by the router manager). Certain log messages (e.g. information on password changes) will not be processed through insecure (SECURE=NO) output definitions to prevent interception by unauthorised parties. If the OUTPUT parameter is set to TEMPORARY, SECURE may not be specified. The default is YES when DESTINATION is set to ROUTER and PASSWORD is set to a valid password, or when DESTINATION is set to MEMORY. For all other cases, the default is NO.

The SERVER parameter specifies a destination IP address for log messages processed by this output definition. The SERVER parameter is required if the DESTINATION parameter is set to ROUTER or SYSLOG, and is not permitted for other values of DESTINATION. When the DESTINATION parameter is set to ROUTER, the SERVER parameter specifies the IP address of the router to transmit SRLP packets to via UDP port 5023. When the DESTINATION parameter is set to SYSLOG, the SERVER parameter specifies the IP address of the Unix host running the syslog server. syslog messages are transmitted via UDP port 514.

The ZONE parameter specifies the time zone to use for time information in log messages, as the recognised name of a time zone (Table 12-7 on page 12-18), or the offset from UTC/GMT of the time zone in which the times should be shown. The default is LOCAL.

Table 12-7: Recognised time zone names.

Time Zone Name	Offset from GMT	Description
ASIA	+8:00	Asia
ACDT	+10:30	Australian Central Daylight Time
ACST	+9:30	Australian Central Standard Time
AEDT	+11:00	Australian Eastern Daylight Time
AEST	+10:00	Australian Eastern Standard Time
AWST	+8:00	Australian Western Standard Time
BST	+1:00	British Standard Time
CHINA	+8:00	China
GMT	+0:00	Greenwich Mean Time
UK	+0:00	Greenwich Mean Time
HK	+8:00	Hong Kong
JST	+9:00	Japan Standard Time
MET	+1:00	Mid-European time
NZDT	+13:00	New Zealand Daylight Time

Table 12-7: Recognised time zone names. (Continued)

Time Zone Name	Offset from GMT	Description
NZST	+12:00	New Zealand Standard Time
SING	+8:00	Singapore
TAIWAN	+8:00	Taiwan
UTC	+0:00	Universal Coordinated Time
CDT	-5:00	US Central Daylight Time
CST	-6:00	US Central Standard Time
EDT	-4:00	US Eastern Daylight Time
EST	-5:00	US Eastern Standard Time
MDT	-6:00	US Mountain Daylight Time
MST	-7:00	US Mountain Standard Time
PDT	-7:00	US Pacific Daylight Time
PST	-8:00	US Pacific Standard Time
DEFAULT	-	-
NONE	-	-

Examples To create an output definition to forward log messages to another router with IP address 192.168.32.7, use the command:

```
CREATE LOG OUTPUT=5 DESTINATION=ROUTER SERVER=192.168.32.7
```

To create an output definition to forward log messages to a local UNIX host with IP address 192.168.32.77, use the command:

```
CREATE LOG OUTPUT=3 DESTINATION=SYSLOG SERVER=192.168.32.77
```

To create an output definition to output log messages to asynchronous port 1 in summary (single-line) format, use the command:

```
CREATE LOG OUTPUT=21 DESTINATION=PORT PORT=1
```

See Also ADD LOG OUTPUT
DELETE LOG OUTPUT
DESTROY LOG OUTPUT
DISABLE LOG OUTPUT
ENABLE LOG OUTPUT
SET LOG OUTPUT

DELETE LOG OUTPUT

Syntax DELETE LOG OUTPUT={TEMPORARY|*output-id*} FILTER={ALL|*filter-id*}

where:

- *output-id* is the index number of an output definition, in the range 1 to 20.

- *filter-id* is a filter entry number, in the range 1 to $n+1$, where n is the number of filters currently defined for the output definition.

Description This command deletes the specified filter entry or entries from the specified output definition.

The OUTPUT parameter specifies the number of the output definition containing the filter to be deleted. The output definition must already exist. If TEMPORARY is specified, the filter is deleted from the special TEMPORARY output definition.

The FILTER parameter specifies the entry number of the filter to be deleted. The filter entry must already exist. If ALL is specified, all filters will be deleted from the specified output definition or definitions.

Examples To delete the first filter entry from output definition 8, use the command:

```
DELETE LOG OUTPUT=8 FILTER=1
```

To delete all filter entries from output definition 10, use the command:

```
DELETE LOG OUTPUT=10 FILTER=ALL
```

See Also ADD LOG OUTPUT
SHOW LOG OUTPUT

DELETE LOG RECEIVE

Syntax DELETE LOG RECEIVE={*ipadd*|ANY}

where:

- *ipadd* is an IP address in dotted decimal notation.

Description This command removes the log receive entry associated with the specified IP address. The RECEIVE parameter specifies the IP address of the entry to be deleted. If ANY is specified, the wildcard entry that matches all IP addresses is removed.

Examples To remove the receive entry for network 192.168.30.0, use the command:

```
DELETE LOG RECEIVE=192.168.30.0
```

See Also ADD LOG RECEIVE
SET LOG RECEIVE
SHOW LOG RECEIVE

DESTROY LOG OUTPUT

Syntax DESTROY LOG OUTPUT={TEMPORARY|*output-id*}

where:

- *output-id* is the index number of an output definition, in the range 1 to 20.

Description This command destroys the specified output definition.

The OUTPUT parameter specifies the index number of the output definition to be destroyed, or the special output definition TEMPORARY. An output definition must already exist with this index number.

Examples To erase output definition 14, use the command:

```
DESTROY LOG OUTPUT=14
```

See Also CREATE LOG OUTPUT
SHOW LOG OUTPUT

DISABLE LOG

Syntax DISABLE LOG

Description This command disables the logging facility, preventing the processing and reception of log messages.

See Also DISABLE LOG GENERATION
DISABLE LOG OUTPUT
DISABLE LOG RECEPTION
ENABLE LOG

DISABLE LOG GENERATION

Syntax DISABLE LOG GENERATION

Description This command disables the generation of log messages on the router. The reception and processing of log messages from other routers is not affected.

See Also DISABLE LOG
DISABLE LOG OUTPUT
DISABLE LOG RECEPTION
ENABLE LOG GENERATION

DISABLE LOG OUTPUT

Syntax DISABLE LOG OUTPUT [= { TEMPORARY | *output-id* }]

where:

- *output-id* is the index number of an output definition, in the range 1 to 20.

Description This command disables the specified output definition. No log messages will be processed by an output definition that is disabled.

The OUTPUT parameter specifies the index number of the output definition that is to be disabled. The specified output definition must exist. If TEMPORARY is specified, the special TEMPORARY output definition is disabled. If no value is specified, log message output definitions 1 to 20 are disabled and will generate no output. The TEMPORARY output definition is not affected.

Examples To disable output definition number 5, use the command:

```
DISABLE LOG OUTPUT=5
```

See Also DISABLE LOG
DISABLE LOG GENERATION
DISABLE LOG RECEPTION
ENABLE LOG OUTPUT

DISABLE LOG RECEPTION

Syntax DISABLE LOG RECEPTION

Description This command disables the reception of log messages from other routers via the Secure Router Logging Protocol (SRLP), the Net Manage Log Protocol and syslog. The generation and processing of local log messages is not affected by this command.

See Also DISABLE LOG
DISABLE LOG GENERATION
DISABLE LOG OUTPUT
ENABLE LOG RECEPTION

ENABLE LOG

Syntax ENABLE LOG

Description This command enables the logging facility. Log messages registered by router modules and received from other routers will now be processed.

See Also DISABLE LOG
ENABLE LOG GENERATION
ENABLE LOG OUTPUT
ENABLE LOG RECEPTION

ENABLE LOG GENERATION

Syntax `ENABLE LOG GENERATION`

Description This command enables the generation of log messages by modules in the router. It does not affect the reception of log messages from other routers over the network. Log message generation can not occur unless the log module itself is enabled.

See Also `DISABLE LOG GENERATION`
`ENABLE LOG`
`ENABLE LOG OUTPUT`
`ENABLE LOG RECEPTION`

ENABLE LOG OUTPUT

Syntax `ENABLE LOG OUTPUT [= {TEMPORARY | output-id}]`

where:

■ *output-id* is the index number of an output definition, in the range 1 to 20.

Description This command enables the specified output definition. An output definition must be enabled before log messages can be processed by the output definition. Output definitions are enabled by default when they are created.

The OUTPUT parameter specifies the index number of the output definition that is to be enabled. The specified output definition must exist. If TEMPORARY is specified, the special TEMPORARY output definition is enabled. If no value is specified, log message output definitions 1 to 20 are enabled and will generate output. The TEMPORARY output definition is not affected.

Examples To enable output definition number 14, use the command:

```
ENABLE LOG OUTPUT=14
```

See Also `DISABLE LOG OUTPUT`
`ENABLE LOG`
`ENABLE LOG GENERATION`
`ENABLE LOG RECEPTION`

ENABLE LOG RECEPTION

Syntax `ENABLE LOG RECEPTION`

Description This command enables the reception of log messages from other routers via the Secure Router Logging Protocol (SRLP), the Net Manage Log Protocol and syslog. Received messages will be processed in the same manner as messages generated on the router. Log messages cannot be received and processed unless the log module is enabled.

See Also DISABLE LOG RECEPTION
ENABLE LOG
ENABLE LOG GENERATION
ENABLE LOG OUTPUT

FLUSH LOG OUTPUT

Syntax FLUSH LOG OUTPUT [= {TEMPORARY | *output-id*}]

where:

■ *output-id* is the index number of an output definition, in the range 1 to 20.

Description This command flushes the queue or queues for the specified output definition or definitions. Flushing an output definition's queue forces the entries in the queue to be processed by the output definition.

The OUTPUT parameter specifies the queue to be flushed. The output definition must already exist. If TEMPORARY is specified, the log messages stored in memory are purged (deleted). If any other output definition is specified, the log messages queued for processing by the specified output definition are processed according to the output definition. If a value is not specified, all queues are flushed.

Examples To force all log messages queued for output definition 3 (which forwards messages to a syslog server) to be forwarded immediately, use the command:

```
FLUSH LOG OUTPUT=3
```

See Also PURGE LOG

PURGE LOG

Syntax PURGE LOG [= {TEMPORARY | *output-id*}]

where:

■ *output-id* is the index number of an output definition, in the range 1 to 20.

Description This command clears the configuration information for the logging facility and/or deletes log messages queued for processing.

If an output definition is not specified and the logging facility is enabled when this command is executed, the configuration is restored to the default state. If the logging facility is disabled, all configuration information is removed from both volatile and nonvolatile storage. All log messages stored in memory are deleted by this command, as are any messages queued for transmission to a router via SRLP or to a syslog server.

If an output definition is specified, only the log message queue for that definition is purged. All messages in it are discarded. Other output definitions are not affected, and the configuration of the logging facility is not altered.

See Also DISABLE LOG
ENABLE LOG

SET LOG OUTPUT

Syntax SET LOG OUTPUT={TEMPORARY|*output-id*} [DESTINATION={MEMORY|PORT|ROUTER|SYSLOG}] [FORMAT={FULL|MSGONLY|SUMMARY}] [MAXQUEUESEVERITY=*severity*] [MESSAGES=*message-count*] [PASSWORD={*password*|NONE}] [PORT=*port-number*] [QUEUEONLY={YES|NO}] [SECURE={YES|NO}] [SERVER=*ipadd*] [ZONE={*time-zone-name*|*utc-offset*}]

SET LOG OUTPUT={TEMPORARY|*output-id*} FILTER=*filter-id* [ACTION={PROCESS|IGNORE}] [ALL] [DATE=[*op*] *dd-mmm-yyyy*] [DEVICE=[*op*] *device*] [FILE=[*op*] *filename*] [MASK=*ipadd*] [MSGTEXT=[*op*] *string*] [MODULE=[*op*] *module-id*] [ORIGIN=*ipadd*] [REFERENCE=[*op*] *string*] [SEVERITY=[*op*] *severity*] [SOURCELINE=[*op*] *line*] [SUBTYPE=[*op*] *subtype-id*] [TIME=[*op*] *hh:mm:ss*] [TYPE=[*op*] *type-id*]

where:

- *output-id* is the index number of an output definition, in the range 1 to 20.
- *severity* is a log message severity, in the range 0 (low) to 7 (high).
- *message-count* is the maximum number of log messages that may be queued for processing.
- *password* is a character string, 1 to 16 characters in length. Valid characters are any printable character. If the string contains spaces it must be enclosed in double quotes.
- *port-number* is the number of an asynchronous port. Ports are numbered sequentially starting with port 0.
- *ipadd* is an IP address in dotted decimal notation.
- *time-zone-name* is the name of a recognised time zone (Table 12-7 on page 12-18).
- *utc-offset* is a time offset from GMT/UTC, in the range +23:59:59 to 23:59:59.
- *filter-id* is a filter entry number, in the range 1 to *n*+1, where *n* is the number of filters currently defined for the output definition.
- *op* is a comparison operator (see Table 12-5 on page 12-9).
- *dd-mmm-yyyy* is a date, where *dd* is the day number (1–31), *mmm* is a three-letter abbreviation for the month (“Jan”, “Feb”, “Mar”, ...) or the month number (1–12), and *yyyy* is the year.
- *device* is a router device number.
- *filename* is a module source file name, 1 to 12 characters in length.
- *ipadd* is an IP address in dotted decimal notation.
- *module-id* is the name or number of a router module (see “Module Identifiers and Names” on page B-2 of Appendix B, Reference Tables for a complete list).
- *string* is a character string, 1 to 15 characters in length.

- *line* is a line number in a module source file, in the range 1 to 65535.
- *subtype-id* is the name or number of a log message subtype (see “Log Message Types and Subtypes” on page B-8 of *Appendix B, Reference Tables* for a complete list).
- *hh:mm:ss* is a time, where *hh* is the hour (0–23), *mm* is the minutes (0–59), and *ss* is the seconds (0–59).
- *type-id* is the name or number of a log message type (see “Log Message Types and Subtypes” on page B-8 of *Appendix B, Reference Tables* for a complete list).

Description This command modifies the specified output definition or log message filter. The output definition specifies the processing to be performed on log messages that match one of the log message filters associated with the output definition. The specified output definition or log message filter must already exist.

The OUTPUT parameter specifies the index number of the output definition to be created, or the special output definition TEMPORARY. If TEMPORARY is specified, the parameters MAXQUEUESEVERITY, QUEUEONLY and SECURE may not be specified. An output definition must already exist with this index number.

The DESTINATION parameter specifies the type of processing to be performed and the destination of log messages processed by this output definition. The MEMORY option stores log messages in RAM. The MEMORY option is only valid only when OUTPUT is set to TEMPORARY. The PORT option outputs log messages to an asynchronous port on the router. If DESTINATION is set to PORT, the parameters MAXQUEUESEVERITY, MESSAGES and QUEUEONLY may not be specified. The ROUTER option forwards log messages via the Secure Router Logging Protocol (SRLP) to another router. The SYSLOG option forwards log messages in syslog format to a syslog server.

The FORMAT parameter specifies the format of log messages when converted into ASCII text for output to an asynchronous port. The FULL option displays the entire log message in multiple lines, with a blank line between messages. The SUMMARY option produces an abbreviated display. The MSGONLY option displays only the text of the message. The FORMAT parameter is valid only if DESTINATION is set to PORT. The default is to display a summary of each log message on a single line, omitting some fields.

The MAXQUEUESEVERITY parameter specifies the maximum message severity level at which messages will be queued but not output, when QUEUEONLY is set to YES. If QUEUEONLY is set to YES, log messages with a severity level less than that specified by the MAXQUEUESEVERITY parameter are queued until the queue length reaches the limit set by the MESSAGES parameter, at which point all the messages are processed. Any log messages with a priority greater than the value of MAXQUEUESEVERITY will cause the queued messages to be flushed (processed). If the DESTINATION parameter is set to PORT or the OUTPUT parameter is set to TEMPORARY, MAXQUEUESEVERITY may not be specified. The default for MAXQUEUESEVERITY is 7 (i.e. only messages with the maximum severity level are output immediately).

The MESSAGES parameter specifies the number of log messages that will be added to the output definition queue before actually being processed. For a DESTINATION of MEMORY, the MESSAGES parameter specifies the maximum number of messages to be stored. When the limit is reached, older messages are purged to make room for new messages. For a DESTINATION of SYSLOG or ROUTER, the MESSAGES parameter specifies the maximum number of messages awaiting processing or acknowledgement. The MESSAGE

parameter is not permitted if DESTINATION is set to PORT. For SYSLOG and ROUTER, the MAXQUEUESEVERITY parameter can cause high-priority messages (and any other queued messages) to be output immediately. The default is 300 for a DESTINATION of RAM, and 20 for a DESTINATION of ROUTER or SYSLOG.

The PASSWORD parameter specifies the password to attach to log messages for authentication purposes when log messages are forwarded to another router via the Secure Router Logging Protocol (SRLP). If the remote router requires a password, this password must match. The PASSWORD parameter is only valid when DESTINATION is set to ROUTER. The password is not transmitted over the network, but is used to compute an MD5 digest. The default is no password.

The PORT parameter specifies an asynchronous port on the router to which log messages are to be directed. The PORT parameter is only valid, and is required, when DESTINATION is set to PORT. If the DESTINATION parameter is set to PORT or the OUTPUT parameter is set to TEMPORARY, QUEUEONLY may not be specified. The default is NO.

The QUEUEONLY parameter controls the output of log messages from the output definition queue. When QUEUEONLY is set to YES log messages are queued by the output definition and are not actually processed (forwarded, printed, displayed, etc.) until the queue is full. The default is NO.

The SECURE parameter specifies whether or not messages processed through this output definition will be “secure” (the meaning of the word “secure” in this context is defined by the router manager). Certain log messages (e.g. information on password changes) will not be processed through insecure (SECURE=NO) output definitions to prevent interception by unauthorised parties. If the OUTPUT parameter is set to TEMPORARY, SECURE may not be specified. The default is YES when DESTINATION is set to ROUTER and PASSWORD is set to a valid password, or when DESTINATION is set to MEMORY. For all other cases, the default is NO.

The SERVER parameter specifies a destination IP address for log messages processed by this output definition. The SERVER parameter is required if the DESTINATION parameter is set to ROUTER or SYSLOG, and is not permitted for other values of DESTINATION. When the DESTINATION parameter is set to ROUTER, the SERVER parameter specifies the IP address of the router to transmit SRLP packets to via UDP port 5023. When the DESTINATION parameter is set to SYSLOG, the SERVER parameter specifies the IP address of the Unix host running the syslog server. syslog messages are transmitted via UDP port 514.

The ZONE parameter specifies the time zone to use for time information in log messages, as the recognised name of a time zone (Table 12-7 on page 12-18), or the offset from UTC/GMT of the time zone in which the times should be shown. The default is LOCAL.

The FILTER parameter specifies the entry number of the filter within the output definition. If FILTER is specified, the filter will be inserted into the filter list at the specified position. If FILTER is not specified, the filter will be added to the end of the filter list for the output definition.

The ACTION parameter specifies the action to perform for log messages matching this filter. If PROCESS is specified, the log message is processed according to the output definition. If IGNORE is specified, the log message is ignored and not processed by this output definition. The default is PROCESS.

The ALL parameter matches all log entries. If ALL is specified, no other selection criteria may be specified for this filter. The default is to only match log entries fitting the specified criteria.

The DATE parameter specifies the date value to match in the log message. The first character of the value may be one of the comparison operators "<", ">" or "!" to modify the comparison from "equals" (the default) to "less than or equal to", "greater than or equal to" or "not equal to" respectively (see Table 12-5 on page 12-9). The default is to match any date.

The DEVICE parameter specifies the device number to match in the log message. The first character of the value may be one of the comparison operators "<", ">" or "!" to modify the comparison from "equals" (the default) to "less than or equal to", "greater than or equal to" or "not equal to" respectively (see Table 12-5 on page 12-9). The default is to match any device number.

The FILE parameter specifies the name of a source file to match in the log message. The first character of the value may be one of the comparison operators "<", ">" or "!" to modify the comparison from "equals" (the default) to "less than or equal to", "greater than or equal to" or "not equal to" respectively (see Table 12-5 on page 12-9). The default is to match any source file.

The MASK parameter specifies a subnet mask to use in association with the ORIGIN IP address parameter. The default is 255.255.255.255.

The MSGTEXT parameter specifies a string to match in the text of the log message. The first character of the value may be one of the comparison operators "<", ">", "!" or "%" to modify the comparison from "equals" (the default) to "less than or equal to", "greater than or equal to", "not equal to" or "contains substring" respectively (see Table 12-5 on page 12-9). The default is to match any text.

The MODULE parameter specifies the router module to match in the log message, as either a decimal number or a recognised module name (see *Module Identifiers and Names* on page B-2 of *Appendix B, Reference Tables* for a complete list). The first character of the value may be one of the comparison operators "<", ">" or "!" to modify the comparison from "equals" (the default) to "less than or equal to", "greater than or equal to" or "not equal to" respectively (see Table 12-5 on page 12-9). The default is to match any module.

The ORIGIN parameter specifies the IP address to match against the originating IP address field of the log message. The MASK parameter can be used to specify a host, subnet or network. The default is to match any IP address in the origin IP address field.

The REFERENCE parameter specifies the reference to match in the log message. The first character of the value may be one of the comparison operators "<", ">", "!" or "%" to modify the comparison from "equals" (the default) to "less than or equal to", "greater than or equal to", "not equal to" or "contains substring" respectively (see Table 12-5 on page 12-9). The default is to match any reference.

The SEVERITY parameter specifies the log message severity level to match in the log message. The first character of the value may be one of the comparison operators "<", ">" or "!" to modify the comparison from "equals" (the default) to "less than or equal to", "greater than or equal to" or "not equal to" respectively (see Table 12-5 on page 12-9). The default is to match any severity.

The SOURCELINE parameter specifies the line number in the source file where the log message was generated, to match in the log message. The first character of the value may be one of the comparison operators "<", ">" or "!" to modify the comparison from "equals" (the default) to "less than or equal to", "greater than or equal to" or "not equal to" respectively (see Table 12-5 on page 12-9). The default is to match any source line number.

The SUBTYPE parameter specifies the log message subtype to match, as either a decimal number or a recognised subtype name (see "Log Message Types and Subtypes" on page B-8 of *Appendix B, Reference Tables* for a complete list). The first character of the value may be one of the comparison operators "<", ">" or "!" to modify the comparison from "equals" (the default) to "less than or equal to", "greater than or equal to" or "not equal to" respectively (see Table 12-5 on page 12-9). The default is to match any log message subtype.

The TIME parameter specifies the time value to match in the log message. The first character of the value may be one of the comparison operators "<", ">" or "!" to modify the comparison from "equals" (the default) to "less than or equal to", "greater than or equal to" or "not equal to" respectively (see Table 12-5 on page 12-9). The default is to match any time.

The TYPE parameter specifies the log message type to match, as either a decimal number or a recognised type name (see "Log Message Types and Subtypes" on page B-8 of *Appendix B, Reference Tables* for a complete list). The first character of the value may be one of the comparison operators "<", ">" or "!" to modify the comparison from "equals" (the default) to "less than or equal to", "greater than or equal to" or "not equal to" respectively (see Table 12-5 on page 12-9). The default is to match any log message type.



Any parameter values that contain spaces must be enclosed in double quotes. If one of the operators ("<", ">", "!", "%") is also present, the operator must be inside the quote marks. For example, SET LOG OUTPUT=4 MSGTEXT="%PPP Inter".

Examples To redirect log messages from output definition 2 to port 4 instead of port 1, use the command:

```
SET LOG OUTPUT=2 PORT=4
```

To change the first filter entry in output definition 2 to ignore rather than process entries, use the command:

```
SET LOG OUTPUT=2 FILTER=1 ACTION=IGNORE
```

See Also CREATE LOG OUTPUT
DESTROY LOG OUTPUT
SHOW LOG OUTPUT

SET LOG RECEIVE

Syntax SET LOG RECEIVE={*ipadd*|ANY} [ALLOW={YES|NO}] [MASK=*ipadd*]
[PASSWORD={*password*|NONE}] [PROTOCOL={ALL|BOTH|NEW|OLD|SYSLOG}]

where:

- *ipadd* is an IP address in dotted decimal notation.

- *password* is a character string, 1 to 16 characters in length. Valid characters are any printable character. If the string contains spaces it must be enclosed in double quotes.

Description This command modifies the options for an entry in the log reception table. The log reception table specifies the routers (and networks/subnets if MASK is specified) from which log messages will be accepted. When the log reception table is empty (the default), log messages will not be accepted from any source. When comparing the source addresses of received log messages, the most specific entry is used. The order of entries in the log reception table is not significant.

The RECEIVE parameter specifies the IP address of the host, subnet or network from which log messages will be received. If ALL is specified, log messages will be accepted from any IP address.

The ALLOW parameter specifies whether or not log messages will be accepted from the specified IP address. If YES is specified, log messages will be accepted from the IP address. If NO is specified, log messages will not be accepted from the IP address. The default is YES.

The MASK parameter specifies a subnet mask to use in association with the RECEIVE parameter. The default is 255.255.255.255 if an IP address is specified for the RECEIVE parameter, or 0.0.0.0 if ALL is specified for the RECEIVE parameter.

The PASSWORD parameter specifies the password that must accompany log messages from the specified IP address, for authentication purposes when log messages are forwarded to another router via the Secure Router Logging Protocol (SRLP). If the PASSWORD option is present, the specified password must accompany log messages from the specified IP address.

The PROTOCOL parameter specifies the protocol to use for message reception from the specified IP address. If OLD is specified the logging facility will accept old Net Manage (UDP port 5024) packets. If NEW is specified the logging facility will accept the new Secure Router Logging Protocol (SRLP) packets. If SYSLOG is specified the logging facility will accept syslog messages. The BOTH option is equivalent to specifying both OLD and NEW. The ALL option is equivalent to specifying OLD, NEW and SYSLOG. The PASSWORD parameter is not valid when TYPE is set to OLD or SYSLOG, as password authentication is not supported in these protocols.

Examples To change the password for router 192.168.37.6, use the command:

```
SET LOG RECEIVE=192.168.37.6 PASSWORD=NEWSECRET
```

See Also ADD LOG RECEIVE
DELETE LOG RECEIVE
SHOW LOG RECEIVE

SET LOG UTCOFFSET

Syntax SET LOG UTCOFFSET={*time-zone-name*|*utc-offset*}

where:

- *time-zone-name* is the name of a recognised time zone (Table 12-7 on page 12-18).
- *utc-offset* is a time offset from GMT/UTC, in the range +23:59:59 to -23:59:59.

Description This command tells the router the difference between local time (the time the router clock is set to) and UTC/GMT time. The router's clock is assumed to be set to local time, so the offset specified by this command is used to calculate UTC time.

The UTCOFFSET parameter specifies the time difference between the router's clock and UTC/GMT, as the recognised name of a time zone (Table 12-7 on page 12-18), or the time difference between the router's clock and UTC/GMT in hours, minutes and seconds. If the router clock is ahead of UTC, this offset is positive.



Although there are some technical differences between the definitions of UTC and GMT time, the router effectively treats UTC and GMT times as equivalent.

Examples To set the UTC offset to +12 hours (appropriate for New Zealand Standard Time), use the command:

```
SET LOG UTCOFFSET=12:00
```

To set the UTC offset to +1 hour (appropriate for British Summer Time), use the command:

```
SET LOG UTCOFFSET=01:00:00
```

To set the UTC offset to zero (appropriate for Greenwich Mean Time and Universal Coordinated Time) use the command:

```
SET LOG UTCOFFSET=0
```

See Also SHOW LOG STATUS

SHOW LOG

Syntax `SHOW LOG [=output-id] [DATE=[op] dd-mm-yyyy]
[DEVICE=[op] device] [FILE=[op] filename] [FULL]
[MASK=ipadd] [MODULE=[op] module-id] [MSGONLY]
[MSGTEXT=[op] string] [ORIGIN=ipadd]
[REFERENCE=[op] string] [REVERSE [=count]]
[SEVERITY=[op] severity] [SOURCELINE=[op] line]
[SUBTYPE=[op] subtype-id] [TAIL [=count]]
[TIME=[op] hh:mm:ss] [TYPE=[op] type-id]
[ZONE={time-zone-name|utc-offset}]`

where:

- *output-id* is the index number of an output definition, in the range 1 to 20.
- *op* is a comparison operator (see Table 12-5 on page 12-9).
- *dd-mm-yyyy* is a date, where *dd* is the day number (1-31), *mmm* is a threeletter abbreviation for the month ("Jan", "Feb", "Mar", ...) or the month number (1-12), and *yyyy* is the year.

- *device* is a router device number.
- *filename* is a module source file name, 1 to 12 characters in length.
- *ipadd* is an IP address in dotted decimal notation.
- *module-id* is the name or number of a router module (see “Module Identifiers and Names” on page B-2 of *Appendix B, Reference Tables* for a complete list).
- *string* is a character string, 1 to 15 characters in length.
- *count* is a number in the range 1 to the number of log messages stored.
- *severity* is a log message severity, in the range 0 (low) to 7 (high).
- *line* is a line number in a module source file, in the range 1 to 65535.
- *subtype-id* is the name or number of a log message subtype (see “Log Message Types and Subtypes” on page B-8 of *Appendix B, Reference Tables* for a complete list).
- *hh:mm:ss* is a time, where *hh* is the hour (0–23), *mm* is the minutes (0–59), and *ss* is the seconds (0–59).
- *type-id* is the name or number of a log message type (see “Log Message Types and Subtypes” on page B-8 of *Appendix B, Reference Tables* for a complete list).
- *time-zone-name* is the name of a recognised time zone (Table 12-7 on page 12-18).
- *utc-offset* is a time offset from GMT/UTC, in the range +23:59:59 to 23:59:59.

Description This command displays the log messages stored in memory (RAM) or in the log message queue for the specified output definition. If an output definition is not specified, the default is to display the log messages stored by the TEMPORARY output definition in RAM. The output can be filtered to display only those entries that match a specific criteria.

The DATE parameter specifies the date value to match in the log message. The first character of the value may be one of the comparison operators “<”, “>” or “!” to modify the comparison from “equals” (the default) to “less than or equal to”, “greater than or equal to” or “not equal to” respectively (see Table 12-5 on page 12-9). The default is to match any date.

The DEVICE parameter specifies the device number to match in the log message. The first character of the value may be one of the comparison operators “<”, “>” or “!” to modify the comparison from “equals” (the default) to “less than or equal to”, “greater than or equal to” or “not equal to” respectively (see Table 12-5 on page 12-9). The default is to match any device number.

The FILE parameter specifies the name of a source file to match in the log message. The first character of the value may be one of the comparison operators “<”, “>” or “!” to modify the comparison from “equals” (the default) to “less than or equal to”, “greater than or equal to” or “not equal to” respectively (see Table 12-5 on page 12-9). The default is to match any source file.

The FULL parameter specifies the format of the display. By default each log message is displayed in summary format on a single line, omitting some fields (Figure 12-2 on page 12-34, Table 12-7 on page 12-18). The FULL parameter displays the entire log message in multiple lines, with a blank line between messages (Figure 12-3 on page 12-36, Table 12-8 on page 12-35).

The MASK parameter specifies a subnet mask to use in association with the ORIGIN IP address parameter. The default is 255.255.255.255.

The MODULE parameter specifies the router module to match in the log message, as either a decimal number or a recognised module name (see *"Module Identifiers and Names"* on page B-2 of *Appendix B, Reference Tables* for a complete list). The first character of the value may be one of the comparison operators "<", ">" or "!" to modify the comparison from "equals" (the default) to "less than or equal to", "greater than or equal to" or "not equal to" respectively (see Table 12-5 on page 12-9). The default is to match any module.

The MSGONLY parameter specifies the format of the display. By default each log message is displayed in summary format on a single line, omitting some fields (Figure 12-2 on page 12-34, Table 12-7 on page 12-18). The MSGONLY parameter displays just the text of the log message.

The MSGTEXT parameter specifies a string to match in the text of the log message. The first character of the value may be one of the comparison operators "<", ">", "!" or "%" to modify the comparison from "equals" (the default) to "less than or equal to", "greater than or equal to", "not equal to" or "contains substring" respectively (see Table 12-5 on page 12-9). The default is to match any text.

The ORIGIN parameter specifies the IP address to match against the originating IP address field of the log message. The MASK parameter can be used to specify a host, subnet or network. The default is to match any IP address in the origin IP address field.

The REFERENCE parameter specifies the reference to match in the log message. The first character of the value may be one of the comparison operators "<", ">", "!" or "%" to modify the comparison from "equals" (the default) to "less than or equal to", "greater than or equal to", "not equal to" or "contains substring" respectively (see Table 12-5 on page 12-9). The default is to match any reference.

The REVERSE parameter specifies that log messages are displayed in reverse date (most recent first) order. If a value is specified, the output is limited to the specified number of log messages. The default, if no value is specified, is to display all log messages.

The SEVERITY parameter specifies the log message severity level to match in the log message. The first character of the value may be one of the comparison operators "<", ">" or "!" to modify the comparison from "equals" (the default) to "less than or equal to", "greater than or equal to" or "not equal to" respectively (see Table 12-5 on page 12-9). The default is to match any severity.

The SOURCELINE parameter specifies the line number in the source file where the log message was generated, to match in the log message. The first character of the value may be one of the comparison operators "<", ">" or "!" to modify the comparison from "equals" (the default) to "less than or equal to", "greater than or equal to" or "not equal to" respectively (see Table 12-5 on page 12-9). The default is to match any source line number.

The SUBTYPE parameter specifies the log message subtype to match, as either a decimal number or a recognised subtype name (see *"Log Message Types and Subtypes"* on page B-8 of *Appendix B, Reference Tables* for a complete list). The first character of the value may be one of the comparison operators "<", ">" or "!" to modify the comparison from "equals" (the default) to "less than or equal to", "greater than or equal to" or "not equal to" respectively (see Table 12-5 on page 12-9). The default is to match any log message subtype.

The TAIL parameter specifies that only the most recent log messages are displayed. If a value is specified, the output is limited to the specified number

of log messages. The default, if no value is specified, is to display the last 20 log messages.

The TIME parameter specifies the time value to match in the log message. The first character of the value may be one of the comparison operators "<", ">" or "!" to modify the comparison from "equals" (the default) to "less than or equal to", "greater than or equal to" or "not equal to" respectively (see Table 12-5 on page 12-9). The default is to match any time.

The TYPE parameter specifies the log message type to match, as either a decimal number or a recognised type name (see "Log Message Types and Subtypes" on page B-8 of *Appendix B, Reference Tables* for a complete list). The first character of the value may be one of the comparison operators "<", ">" or "!" to modify the comparison from "equals" (the default) to "less than or equal to", "greater than or equal to" or "not equal to" respectively (see Table 12-5 on page 12-9). The default is to match any log message type.

The ZONE parameter specifies the time zone to use for time information in log messages, as the recognised name of a time zone (Table 12-7 on page 12-18), or the offset from UTC/GMT of the time zone in which the times should be shown. The default is LOCAL.



Although there are some technical differences between the definitions of UTC and GMT time, the router effectively treats UTC and GMT times as equivalent.

Figure 12-2: Example output from the SHOW LOG command.

```

Date/Time   S Mod  Type  SType Message
-----
17 10:22:37 2 PPP  ATT   ATTCH ppp0: The IP module has attached
17 10:22:37 2 PPP  ATT   ATTCH ppp1: The IP module has attached
17 10:22:37 7 SYS  REST  NORM  Router startup, version 7.2-00, 21-Jun-1996,
4096k RAM
17 10:22:38 2 PPP  DLINK UP    ppp0: Primary link over syn0 has opened
17 10:22:38 3 PPP  VINT  UP    ppp0: Interface has come up and is able to send
and receive data
17 10:22:38 2 PPP  CIRC  UP    ppp0: IPCP has opened
17 10:31:28 2 CH   CMD   MGR   show log
17 10:36:43 2 CH   CMD   MGR   show log zone=-24:00
17 10:36:43 4 CH   MSG   ERROR Illegal time: 24:00
17 10:43:57 0 LOG                IP, telnet connection accepted from
202.36.163.20
17 10:44:00 3 USER USER   LON   manager login on TTY18
17 10:44:21 2 CH   CMD   MGR   show loge status
17 10:44:21 4 CH   MSG   ERROR Parameter "loge" not recognised
17 10:44:24 2 CH   CMD   MGR   show log status
17 10:44:50 2 CH   CMD   MGR   lo
17 10:44:50 3 USER USER   LOFF  manager logoff on TTY18
17 10:44:52 2 CH   CMD   MGR   show log
17 10:45:41 0 LOG                IP, telnet connection accepted from
202.36.163.20
17 10:45:54 3 USER USER   LON   manager login on TTY18
17 10:45:58 2 CH   CMD   MGR   show log

```


Table 12-8: Parameters displayed in the output of the SHOW LOG command.

Parameter	Meaning
Date/Time	The date and time the log message was generated. The date is displayed as just the day number (1–31).
S	The severity of the log message.
Mod	The name of the module that generated the log message (see “ <i>Module Identifiers and Names</i> ” on page B-2 of <i>Appendix B, Reference Tables</i> for a complete list).
Type	The message type (see “ <i>Log Message Types and Subtypes</i> ” on page B-8 of <i>Appendix B, Reference Tables</i> for a complete list).
SType	The message subtype (see “ <i>Log Message Types and Subtypes</i> ” on page B-8 of <i>Appendix B, Reference Tables</i> for a complete list).
Message	The contents of the <i>Message</i> field in the log message. For log messages of type IPFILT/PASS, the format of the message text is “ <i>filter-number/entry-number Pass Fail src-ipadd>dest-ipadd protocol src-port>dest-port packet-size:data-size</i> ”. For log messages of type IPFILT/DUMP, the message text contains the first 32 octets of the packet.

Figure 12-3: Example output from the SHOW LOG FULL command.

Date/Time	Mod	Type	SType	Dev	Origin	MSGID	Source File/Line
15:38:47 03-JUN-1997	7 00400040	SYS REST	NORM	00000	Local	00001	shostart.c:179
Router startup, version 7.4-00, 12-May-1997, Clock Log : 15:38:33 on 03-Jun-1997							
15:38:48 03-JUN-1997	3 ppp0	PPP VINT	UP	00000	Local	00030	pppinter.c:1655
ppp0: Interface has come up and is able to send and receive data							
15:40:47 03-JUN-1997	5 ppp0	PPP DLINK	ERROR	00000	Local	00032	ppplqm.c:476
ppp0: An LQR failure has occurred on primary link over syn0							
15:40:47 03-JUN-1997	5 ppp0	PPP DLINK	DOWN	00000	Local	00033	ppplcp.c:1017
ppp0: Primary link over syn0 has closed							
15:40:47 03-JUN-1997	5 ppp0	PPP VINT	DOWN	00000	Local	00034	pppinter.c:1792
ppp0: Interface has gone down and is unable to send or receive data							
15:40:50 03-JUN-1997	3 ppp0	PPP VINT	UP	00000	Local	00037	pppinter.c:1655
ppp0: Interface has come up and is able to send and receive data							
10:04:06 04-JUN-1997	4 331012	CH MSG	ERROR	00016	Local	00058	utlmsg.c:930
Parameter "tot" not recognised							
10:04:36 04-JUN-1997	4 331012	CH MSG	ERROR	00016	Local	00061	utlmsg.c:930
Parameter "tot" not recognised							
10:04:42 04-JUN-1997	4 331012	CH MSG	ERROR	00016	Local	00063	utlmsg.c:930
Parameter "totta" not recognised							
10:04:45 04-JUN-1997	4 331259	CH MSG	ERROR	00016	Local	00065	utlmsg.c:930
Invalid parameter combination. The correct command should be: SHOW FL[ASH] {P[H							
10:05:47 04-JUN-1997	4 305012	CH MSG	ERROR	00016	Local	00070	utlmsg.c:930
Parameter "log" not recognised							

Table 12-9: Parameters displayed in the output of the SHOW LOG FULL command.

Parameter	Meaning
Date/Time	The date and time the log message was generated, including the UTC offset.
S	The severity of the log message.
Mod	The name of the module that generated the log message (see <i>"Module Identifiers and Names"</i> on page B-2 of <i>Appendix B, Reference Tables</i> for a complete list).
Type	The message type (see <i>"Log Message Types and Subtypes"</i> on page B-8 of <i>Appendix B, Reference Tables</i> for a complete list).
SType	The message subtype (see <i>"Log Message Types and Subtypes"</i> on page B-8 of <i>Appendix B, Reference Tables</i> for a complete list).
Dev	The device (e.g. asynchronous port or TTY session) that triggered the log message.
Origin	The origin of the log message; one of "Local" or the IP address of the host which generated the log message, via either SRLP or syslog.
MSGID	The message ID number.
Source File/Line	The file name and line number of the module source file where the log message originated.
Ref	The contents of the <i>Reference</i> field in the log message.
Flags	The contents of the <i>Flags</i> field in the log message; one or more of "LOCTIME", "SECURE" or "CMDOUT".
Message	The contents of the <i>Message</i> field in the log message.

Examples To display all recent log messages, use the command:

```
SHOW LOG
```

To display log messages for critical events, use the command:

```
SHOW LOG SEVERITY=>8
```

To display log messages for recent user activity, use the command:

```
SHOW LOG TYPE=USER
```

The PPP link to Head Office is not working. To display log messages relating to link activity, use the command:

```
SHOW LOG TYPE=LINK
```

A terminal connected to port 0 and logged in with MANAGER privilege was left unattended. To see if someone has interfered with the router, use the command:

```
SHOW LOG TYPE=CMD DEVICE=PORT0
```

There is a problem with one of the modems. To see who has been affected, use the command:

```
SHOW LOG TYPE=USER SUBTYPE=LOGIN DEVICE=PORT4
```

See Also PURGE LOG
SHOW LOG STATUS

SHOW LOG COUNTERS

Syntax SHOW LOG COUNTERS

Description This command displays diagnostic counters for the logging facility (Figure 12-4 on page 12-38, Table 12-9 on page 12-37).

Figure 12-4: Example output from the SHOW LOG COUNTERS command.

```
Log Counters

Idle loop passes ..... 283
Transmit passes ..... 78

Messages Generated ..... 136

Messages Received (Syslog) ..... 0
Messages Received (Old protocol) ..... 0
Messages Received (New protocol, SRLP) ..... 0

Messages Rejected (Syslog) ..... 0
Messages Rejected (Old protocol) ..... 0
Messages Rejected (New protocol, SRLP) ..... 0
Messages Rejected (Module disabled) ..... 0
Messages Rejected (Generation disabled) ..... 0
Messages Rejected (Reception disabled) ..... 0
Messages Rejected (Bad parameters) ..... 0

Messages with invalid time ..... 0

Messages Transmitted (Syslog) ..... 0
Messages Transmitted (New protocol, SRLP) ..... 72

Messages Retransmitted (New protocol, SRLP) ..... 0
ACKs Sent (New protocol) ..... 0
ACKs Sent (Old protocol) ..... 0
ACKs Received (New protocol, SRLP) ..... 72

Message transmissions failed (New protocol, SRLP) ..... 0

Messages processed via OD 1 ..... 78 (Router)
Messages processed via OD TE ..... 21 (Memory)
```

Table 12-10: Parameters displayed in the output of the SHOW LOG COUNTERS command.

Parameter	Meaning
Idle loop passes	The number of times the log message handling process has been activated from the router idle loop.
Transmit passes	The number of times the log message transmission process has been activated.
Messages Generated	The number of log messages generated on this router.
Messages Received (Syslog)	The number of log messages received via syslog by this router.
Messages Received (Old protocol)	The number of log messages received via the Net Manage logging protocol by this router.

Table 12-10: Parameters displayed in the output of the SHOW LOG COUNTERS command. (Continued)

Parameter	Meaning
Messages Received (New protocol...	The number of log messages received via the Secure Router Log Protocol (SRLP) by this router.
Messages Rejected (Syslog)	The number of log messages received via syslog by this router that were rejected.
Messages Rejected (Old protocol)	The number of log messages received via the Net Manage logging protocol by this router that were rejected.
Messages Rejected (New protocol...	The number of log messages received via the Secure Router Log Protocol (SRLP) by this router that were rejected.
Messages Rejected (Module...	The number of log messages received by this router that were rejected because the logging facility is disabled on this router.
Messages Rejected (Generation...	The number of log messages from software modules on this router that were rejected because log message generation is disabled on this router.
Messages Rejected (Reception...	The number of log messages received by this router that were rejected because log message reception is disabled on this router.
Messages Rejected (Bad...	The number of log messages received by this router that were rejected because they contained invalid parameter values.
Messages with invalid time	The number of messages with an invalid timestamp.
Messages Transmitted (Syslog)	The number of log messages transmitted via syslog by this router.
Messages Transmitted (New...	The number of log messages transmitted via the Secure Router Log Protocol (SRLP) by this router.
Messages Retransmitted...	The number of log messages retransmitted via the Secure Router Log Protocol (SRLP) by this router.
ACKs Sent (New protocol)	The number of acknowledgements transmitted for log messages received via the Secure Router Log Protocol (SRLP) by this router.
ACKs Sent (Old protocol)	The number of acknowledgements transmitted for log messages received via the Net Manage logging protocol by this router.
ACKs Received (New protocol...	The number of acknowledgements received for log messages transmitted via the Secure Router Log Protocol (SRLP) by this router.
Message transmissions failed...	The number of retransmissions of log messages via the Secure Router Log Protocol (SRLP) that have failed.
Messages processed via OD <i>n (type)</i>	The number of messages processed by the specified output definition.

Examples To display diagnostic counters, use the command:

```
SHOW LOG COUNTERS
```

See Also SHOW LOG
SHOW LOG OUTPUT
SHOW LOG STATUS
SHOW LOG QUEUE

SHOW LOG OUTPUT

Syntax SHOW LOG OUTPUT [= {TEMPORARY | *output-id*}]
[{FILTER=*filter-id* | FULL}]

where:

- *output-id* is the index number of an output definition, in the range 1 to 20.
- *filter-id* is a filter entry number, in the range 1 to *n*+1, where *n* is the number of filters currently defined for the output definition.

Description This command displays the specified or all output definitions. If a value is not specified for the OUTPUT parameter, and neither the FILTER or FULL parameters are specified, the default is to display summary details of all output definitions (Figure 12-5 on page 12-40, Table 12-10 on page 12-38).

The OUTPUT parameter specifies the index number of the output definition to be displayed, or the special output definition TEMPORARY. If a value is not specified, the default is to display all output definitions.

If FULL is specified, detailed information about each output definition including details of all message filters is displayed (Figure 12-6 on page 12-42, Table 12-11 on page 12-41). The FILTER and FULL parameters are mutually exclusive — only one may be specified in any one command.

The FILTER parameter produces the same output as the FULL parameter, except that detailed filter information is only displayed for the specified filter. The FILTER and FULL parameters are mutually exclusive — only one may be specified in any one command.

Figure 12-5: Example output from the SHOW LOG OUTPUT command.

OD#	Type	Port	Server	Msg	Zone	Fmt	ESQMP
01	Syslog		202.36.163.20		----		YNN--
02	Router		202.36.163.40		----		YNN--
		TE	Memory			0200 ----	YY----

Table 12-11: Parameters displayed in the output of the SHOW LOG OUTPUT command.

Parameter	Meaning
OD#	The index number of the output definition.
Type	The destination for log messages processed by this output definition; one of "Memory", "Port", "Router" or "Syslog".
Port	The asynchronous port number on the router to which log messages will be directed by this output definition, when the <i>Type</i> field is "Port".
Server	The IP address of the router or host to which log messages will be directed by this output definition, when the <i>Type</i> field is "Router" or "Syslog".
Msg	The maximum number of messages that may be queued for processing by this output definition.
Zone	The time zone in which date and time values will be displayed and processed by this output definition; one of "Local", "GMT", "UTC", "-", (not set), or an offset from UTC in the range -23:59:59 to +23:59:59.
Fmt	The format of log messages processed by this output definition; one of "Full" or "Summary".
ESQMP	The values of the ENABLED, SECURE, QUEUEONLY, MAXQUEUESEVERITY, and PASSWORD parameters for this output definition. For ENABLED, SECURE and QUEUEONLY the value is one of "Y" (Yes), "N" (No) or "-" (not applicable). For MAXQUEUESEVERITY the value is a severity level in the range 0 to 7. For PASSWORD the value is "-" (password not set) or "*" (password set).

Figure 12-6: Example output from the SHOW LOG OUTPUT FULL command.

```

Output Definition ..... 1
Enabled ..... Yes
Type ..... Syslog
IP Address (Server) ..... 202.36.163.20
Time Zone ..... -
Secure ..... No
Queue Only ..... No

Output Definition ..... 2
Enabled ..... Yes
Type ..... Router
IP Address (Server) ..... 202.36.163.40
Time Zone ..... -
Secure ..... No
Queue Only ..... No

Filter 1:
  MODULE != IP
  SEVERITY < 7
  ---> Process
Filter 2:
  ALL

Output Definition ..... Temporary
Enabled ..... Yes
Type ..... Memory
Max Messages ..... 200
Time Zone ..... -
Secure ..... Yes

Filter 1:
  ALL

```

Table 12-12: Parameters displayed in the output of the SHOW LOG OUTPUT FULL command.

Parameter	Meaning
Output Definition	The index number of the output definition, or TE (Temporary).
Enabled	Whether or not the output definition is enabled and will process log messages matching any of the associated filters; one of "Enabled" or "Disabled".
Type	The destination for log messages processed by this output definition; one of "Memory", "Port", "Router" or "Syslog".
IP Address (Server)	The IP address of the router or host to which log messages will be directed by this output definition, when the <i>Type</i> field is "Router" or "Syslog".
Zone	The time zone in which date and time values will be displayed and processed by this output definition, displayed as an offset from UTC in the range -23:59:59 to +23:59:59, followed by the abbreviation for the time zone (if defined).
Secure	Whether or not log messages processed by this output definition will be "secure"; one of "Yes" or "No".

Table 12-12: Parameters displayed in the output of the SHOW LOG OUTPUT FULL command. (Continued)

Parameter	Meaning
Queue Only	Whether or not log messages matching one of the filters associated with this output definition will be just queued for processing, or processed; one of "Yes" or "No".
Max Messages	The maximum number of messages that may be queued for processing by this output definition.
Filter #	The index of an associated message filter, the filter attributes to match, and the action to perform. The filter attributes may be "ALL" (matches all messages) or one or more conditions. The action is one of "---> Process" or "---> Ignore".
Port	The asynchronous port number on the router to which log messages will be directed by this output definition, when the <i>Type</i> field is "Port".
Format	The format of log messages processed by this output definition; one of "Full" or "Summary".
Password	The password attached to log messages for authentication purposes if log messages are to be forwarded to another router via SRLP by this output definition, or "NONE" if a password has not been set.
Max Queue Severity	The maximum severity level at which log messages are queued and not processed immediately by this output definition; in the range 0 (low) to 7 (high).

Examples To display all output definitions, use the command:

```
SHOW LOG OUTPUT
```

To display only output definition number 7, use the command:

```
SHOW LOG OUTPUT=7
```

To display the filter entries for output definition 7, use the command:

```
SHOW LOG OUTPUT=7 FULL
```

To display the second filter entry for output definition 7, use the command:

```
SHOW LOG OUTPUT=7 FILTER=2
```

See Also ADD LOG OUTPUT
CREATE LOG OUTPUT
DELETE LOG OUTPUT
DESTROY LOG OUTPUT
SET LOG OUTPUT
SHOW LOG STATUS

SHOW LOG QUEUE

Syntax `SHOW LOG QUEUE [=output-id]`

where:

- *output-id* is the index number of an output definition, in the range 1 to 20.

Description This command displays information about the messages in log message queues and the messages that have been transmitted via SRLP but are awaiting acknowledgement. If an output definition is specified, only the entries in the log message queues for the specified definition are displayed. If an output definition is not specified, all entries in the log message queues are displayed (Figure 12-7 on page 12-44, Table 12-12 on page 12-42).

Figure 12-7: Example output from the SHOW LOG QUEUE command.

Queue	RAM Messages	NVS Messages	Type
01	0000/0100	0007/0010	Router
TE	0021/0200	0000/0000	Memory

Outstanding SRLP Messages (Sent but not acknowledged)

OD#	Message ID	Last Attempt	Attempts	Delay
01	73	0	0	0
01	74	0	0	0
01	75	0	0	0
01	76	0	0	0
01	77	0	0	0
01	78	0	0	0
01	79	0	0	0

Table 12-13: Parameters displayed in the output of the SHOW LOG QUEUE command.

Parameter	Meaning
Queue	The output definition with which this queue is associated; one of "TE" (TEMPORARY), or an output definition identifier in the range 1 to 20.
RAM Messages	The number of messages currently stored in RAM and the maximum number of messages that may be stored in RAM.
Type	The destination for log messages in this queue; one of "Memory", "Port", "Router" or "Syslog".
OD#	The index number of an output definition.
Message ID	The message ID number
Last Attempt	The time, expressed as seconds since midnight, that the last attempt was made to retransmit the message.
Attempts	The number of attempts made to retransmit the message.
Delay	The delay, in seconds, between each retransmission.

Examples To display the entries in the log message queue for output definition 3, use the command:

```
SHOW LOG QUEUE=3
```

To display information about all log message queues, use the command:

```
SHOW LOG QUEUE
```

See Also SHOW LOG
SHOW LOG OUTPUT
SHOW LOG STATUS

SHOW LOG RECEIVE

Syntax SHOW LOG RECEIVE [=ipadd] [MASK=ipadd]

where:

- *ipadd* is an IP address in dotted decimal notation.

Description This command displays entries from the log reception table (Figure 12-8 on page 12-45, Table 12-13 on page 12-44). If an IP address is supplied, only the entry for that address is displayed. If a network mask is also supplied any entries within the subnet defined by the IP address and network mask are displayed.

The RECEIVE parameter specifies the IP address to display. If an IP address is not specified, all entries in the log reception table are displayed.

The MASK parameter specifies a subnet mask to use in association with the RECEIVE parameter. The default is 255.255.255.255 if an IP address is specified for the RECEIVE parameter, or 0.0.0.0 if ALL is specified for the RECEIVE parameter. If MASK is specified, all entries falling in the range of IP addresses specified by the combination of RECEIVE and MASK are displayed.

Figure 12-8: Example output from the SHOW LOG RECEIVE command.

Type	IP/Network Addr	Netmask	Protocol	Password
Allow	192.168.0.0	255.255.0.0	BOTH	—
Allow	192.168.2.0	255.255.255.0	NEW	*****

Table 12-14: Parameters displayed in the output of the SHOW LOG RECEIVE command.

Parameter	Meaning
Allow	Whether or not messages will be received from the IP address; one of "Allow" or "Reject".
IP/Network Addr	The IP address of a host, subnet or network from which log messages will be received, or "Any" if log messages will be received from any IP address.
Netmask	The subnet mask to use in association with the IP address.
Protocol	The type of message that will be received from the IP address; one of "Old" (old Net Manage messages), "New" (new format messages), "Both" (equivalent to Old + New), "Syslog" or "Any" (equivalent to Old + New + Syslog).
Password	The password that must accompany messages from the IP address, or "-" if a password is not required.

Examples To display the entry for router 192.168.1.11, use the command:

```
SHOW LOG RECEIVE=192.168.1.11
```

To display all entries, use the command:

```
SHOW LOG RECEIVE
```

To display all entries for routers in subnet 192.168.40.0, use the command:

```
SHOW LOG RECEIVE=192.168.40.0 MASK=255.255.255.0
```

See Also ADD LOG RECEIVE
DELETE LOG RECEIVE
SET LOG RECEIVE
SHOW LOG STATUS

SHOW LOG STATUS

Syntax SHOW LOG STATUS

Description This command displays configuration information for the logging facility (Figure 12-9 on page 12-46, Table 12-14 on page 12-46).

Figure 12-9: Example output from the SHOW LOG STATUS command.

```
Log System Status
-----

Log Module Status ..... Enabled
Log Message Generation ..... Enabled
Log Message Reception (via network) ... Enabled
Log Message Output ..... Enabled
Local Time Offset (from UTC) ..... 12:00:00 (NZST)
Next Message ID ..... 12
Number of Output Definitions ..... 2
```

Table 12-15: Parameters displayed in the output of the SHOW LOG STATUS command.

Parameter	Meaning
Log Module Status	The current status of the logging facility; one of "Enabled" or "Disabled".
Log Message Generation	Whether or not log messages will be generated by modules in this router; one of "Enabled" or "Disabled".
Log Message Reception	Whether or not log messages will be received from the network by this router; one of "Enabled" or "Disabled".
Log Message Output	Whether or not any output definitions are enabled and will generate output; one of "Enabled" or "Disabled".
Local Time Offset (from UTC)	The offset of local time from UTC time, in the range +23:59:59 to -23:59:59, or "-" if a UTC offset has not been set.
Next Message ID	The unique message identifier that will be assigned to the next log message processed by the logging facility.
Number of Output Definitions	The number of output definitions currently defined.

Chapter 13

Scripting

Introduction	13-2
Creating Scripts	13-2
Script Commands	13-2
Using the Built-in Text Editor	13-3
Loading from a TFTP Server	13-3
Loading from an Asynchronous Port	13-3
Using Scripts	13-4
Script Parameters	13-4
Script Control Structures	13-4
Command Reference	13-5
ACTIVATE SCRIPT	13-5
ADD SCRIPT	13-6
DEACTIVATE SCRIPT	13-7
DELETE SCRIPT	13-8
IF..THEN..ELSE..ENDIF	13-8
SET SCRIPT	13-10
SHOW SCRIPT	13-11
WAIT	13-12

Introduction

This chapter describes the scripting facility provided by the router, and how to create and run scripts.

The router's command processor accepts configuration commands entered from a terminal connected to an asynchronous port or a Telnet connection. The command line editing and recall functions enable previous commands to be recalled, edited and re-executed. However, this approach can be cumbersome if many similar commands have to be entered, or if sequences of commands have to be entered repeatedly at different times or on different routers.

The scripting facility enables sequences of commands to be stored in a script, and replayed at any time, allowing the router to be easily configured or quickly re-configured. Scripts can be activated either from the command line, using the `ACTIVATE SCRIPT` command on page 13-5, or from a trigger.

Scripts are stored in the router's file system as text files in FLASH. By convention, scripts with the file type "CFG" contain configuration commands to be executed at boot. Scripts with the file type "SCP" are intended for other repetitive tasks. A special configuration script `BOOT.CFG`, if found in FLASH, will be executed at boot. This script allows a sequence of commands to be executed every time the router reboots.

Creating Scripts

Scripts are text files containing standard router configuration commands that would normally be entered at the router's command line prompt. A script can be created using any one of the following methods:

- Using script commands entered at the router's command line prompt.
- Using the router's built-in text editor.
- Loading the file from a TFTP server using the `LOAD` command.
- Loading the file over from asynchronous port using the `LOAD` command.

Script Commands

A script can be created from the command line, using the command:

```
ADD SCRIPT=filename [LINE=line] [TEXT=text]
```

Additional lines can be added to the script by repeating the command as often as required. The text of a line in a script file can be changed using the command:

```
SET SCRIPT=filename LINE=line TEXT=text
```

The lines in a script file can be re-ordered using either of the commands:

```
SET SCRIPT=filename LINE=line BEFORE=line  
SET SCRIPT=filename LINE=line AFTER=line
```

The contents of a script file can be displayed using the command:

```
SHOW SCRIPT[=filename]
```


The use of commands to create scripts is rather cumbersome and is not recommended, unless the script is very short or none of the other methods can be used. It is much easier to use the router's built-in text editor, or to create the file on a PC and download it using the LOAD command.

Using the Built-in Text Editor

The router's built-in text editor can be used to create scripts. The editor is invoked using the command:

```
EDIT [filename]
```

The editor uses VT100 command sequences and should only be used with a VT100-compatible terminal, terminal emulation program or Telnet client. Scripts created using the editor must be named with a file type of "SCP" or "CFG" so they can be identified correctly by the system.



Before starting the editor make sure your terminal, terminal emulation program or Telnet client is 100% compatible with a VT100 terminal.

See *Chapter 1, Operation* for more information about using the built-in editor.

Loading from a TFTP Server

Script files can be downloaded from a TFTP server using the command:

```
LOAD [FILE=filename] [DESTINATION=FLASH] [SERVER=ipadd]  
[DELAY=delay]
```

The advantages of loading script files from a TFTP server are that the files can be created using any available plain text editor or application that generates plain ASCII text files, scripts can be shared and used on any number of routers, and the scripts for an entire network of routers can be managed centrally under change control if required.

Loading from an Asynchronous Port

Script files can be downloaded over one of the router's asynchronous ports using the command:

```
LOAD [FILE=filename] [DESTINATION=FLASH] [PORT=port]  
[DELAY=delay]
```

After the LOAD command is executed, all input received via the specified asynchronous port is captured and saved in the specified file. The load stops when a control character other than a carriage return (ASCII 13) or line feed (ASCII 10) is received.

Using Scripts

Script can in themselves activate other scripts. The newly activated child script is independent of the parent script. The two scripts will run in parallel.



As scripts can activate other scripts, great care should be taken not to make a loop of script activation.

To minimise the impact on the system of executing a script, a brief pause is inserted between the execution of each line of a script. The only exception to this is the boot script, BOOT.CFG, which executes commands with no delays.

The output from a script can be directed to either the TTY device (a terminal connected to an asynchronous port or a Telnet connection) from which the script was activated, or to the logging facility. The default is to direct all output from the boot script BOOT.CFG to the logging facility, and all output from other scripts to the TTY device.

Script Parameters

Up to eight parameters can be passed to a script. Parameters are specified on the command line after the script name, separated by spaces:

```
ACTIVATE SCRIPT=[filename] [param1 param2 param3 param4
                           param5 param6 param7 param8]
```

Within a script the symbols %1 to %8 are used to refer to the passed parameters, and are automatically replaced by the parameter values before the script is executed. Parameters allow more generic scripts to be written to handle certain operations.

Script Control Structures

The IF..THEN..ELSE..ENDIF control structure can be used execute a different set of router commands depending on some condition:

```
IF string1 {EQ|NE} string2 THEN
    router commands...
ENDIF

IF string1 {EQ|NE} string2 THEN
    router commands...
ELSE
    router commands...
ENDIF
```

The EQ and NE logical operators test that *string1* and *string2* are equal or not equal, respectively. Tests are not case sensitive, so the following expressions are equivalent:

```
FLASH EQ FLASH
FLASH EQ flash
```

If the result of the expression is true, then the router commands between the `IF . . THEN` and `ELSE` or `ENDIF` statements are executed. Control continues with the next statement after the `IF . . THEN . . ELSE . . ENDIF` statement. If the result of the expression is false and there is an `ELSE` clause, then the router commands between the `ELSE` and `ENDIF` statements are executed. Control continues with the next statement after the `IF . . THEN . . ELSE . . ENDIF` statement. If the result of the expression is false and there is no `ELSE` clause, then control continues with the next statement after the `IF . . THEN . . ELSE . . ENDIF` statement.

By using parameters in conjunction with `IF . . THEN . . ELSE . . ENDIF` control structures, a script can be written to behave differently depending on the values of the parameters passed to the script. For example, consider the following script, called `L.SCP`:

```
LOAD FILE=%1 DEST=FLASH SERVER=202.36.163.10
IF %2 EQ GO THEN
    ACTIVATE SCRIPT=%1 %3 %4 %5 %6 %7 %8
ENDIF
```

The script can be activated to load a file into FLASH with the command:

```
ACTIVATE SCRIPT=L.SCP LINKUP.SCP GO PPP0
```

Command Reference

This section describes the commands available on the router to configure the script facility, and to create and execute scripts.

See “Conventions” on page xxxv of *Preface* at the front of this manual for details of the conventions used to describe command syntax. See *Appendix A, Messages* for a complete list of all messages and their meanings.

ACTIVATE SCRIPT

Syntax `ACTIVATE SCRIPT=filename [OUTPUT=device] [parameters]`

where:

- *filename* is a file name of the form `device:filename.type`. *device* is the name of the memory device in which the file is stored (e.g. FLASH). *type* must be “SCP” or “CFG”. Valid characters are uppercase letters (A–Z), lowercase letters (a–z), digits (0–9) and the hyphen character (-). Wildcards are not allowed.
- *device* is the name of the device to which the output from the script will be directed (e.g. LOG).
- *parameters* is a list of one to eight parameters. Each parameter is a character string, 1 to 255 characters in length. Valid characters are any printable character.

Description This command activates the playing of a script file. The `SCRIPT` parameter specifies the file name of the script. A complete file name must be specified, including device, filename and type. The type must be “SCP” or “CFG”.

The OUTPUT parameter specifies the name of the device to which the output from the script will be directed. The only output device currently supported is the logging facility (LOG). If OUTPUT is not specified, the output from the script is sent to the TTY device (a terminal connected to an asynchronous port or a Telnet connection) from which the script was activated.

Up to eight parameters can be passed to a script. Parameters are specified on the command line after the script name, separated by spaces. Within the script, the parameters are referenced by the symbols %1 to %8, which are replaced at run time by the parameter values.



For security reasons this command will only be accepted if the user has SECURITY OFFICER privilege.

Examples To activate a script called SHOWME.SCP, use the command:

```
ACTIVATE SCRIPT=SHOWME.SCP
```

See Also ADD SCRIPT
DELETE SCRIPT
DEACTIVATE SCRIPT
SET SCRIPT
SHOW SCRIPT

ADD SCRIPT

Syntax ADD SCRIPT=*filename* TEXT=*text* [LINE=*line*]

where:

- *filename* is a file name of the form `device:filename.type`. *device* is the name of the memory device in which the file is stored (e.g. FLASH). *type* must be "SCP" or "CFG". Valid characters are uppercase letters (A–Z), lowercase letters (a–z), digits (0–9) and the hyphen character (-). Wildcards are not allowed.
- *text* is a character string, 1 to 127 characters in length. Valid characters are any printable character. If the string contains spaces it must be enclosed in double quotes.
- *line* is the number of a line in the script, expressed as a decimal number.

Description This command adds a line of text to an existing script. The SCRIPT parameter specifies the file name of the script. A complete file name must be specified, including device, filename and type. The type must be "SCP" or "CFG".

The TEXT parameter specifies the line of text to add to the script.

The LINE parameter specifies the line in the script after which the new line of text will be inserted. If the LINE parameter is not specified, the new line of text will be added to the end of the script.



For security reasons this command will only be accepted if the user has SECURITY OFFICER privilege.

Examples To add a script called SHOWME.SCP, use the command:

```
ADD SCRIPT=SHOWME.SCP TEXT="SHOW LOG"
```

See Also ACTIVATE SCRIPT
DELETE SCRIPT
DEACTIVATE SCRIPT
SET SCRIPT
SHOW SCRIPT
WAIT

DEACTIVATE SCRIPT

Syntax DEACTIVATE SCRIPT=*filename*

where:

- *filename* is a file name of the form `device:filename.type`. `device` is the name of the memory device in which the file is stored (e.g. FLASH). `type` must be "SCP" or "CFG". Valid characters are uppercase letters (A–Z), lowercase letters (a–z), digits (0–9) and the hyphen character (-). Wildcards are not allowed.

Description This command stops the playing of a script file.

The SCRIPT parameter specifies the file name of the script. A complete file name must be specified, including device, filename and type. The type must be "SCP" or "CFG". Because of the speed that scripts play, and their generally small size, it may not be practical to stop a script once it has been activated.



For security reasons this command will only be accepted if the user has SECURITY OFFICER privilege.

Examples To deactivate a script called SHOWME.SCP, use the command:

```
DEACTIVATE SCRIPT=SHOWME.SCP
```

See Also ACTIVATE SCRIPT
ADD SCRIPT
DELETE SCRIPT
SET SCRIPT
SHOW SCRIPT

DELETE SCRIPT

Syntax `DELETE SCRIPT=filename [LINE=line]`

where:

- *filename* is a file name of the form `device:filename.type`. *device* is the name of the memory device in which the file is stored (e.g. FLASH). *type* must be "SCP" or "CFG". Valid characters are uppercase letters (A–Z), lowercase letters (a–z), digits (0–9) and the hyphen character (-). Wildcards are not allowed.
- *line* is the number of a line in the script, expressed as a decimal number.

Description This command deletes an entire script file, or deletes a line of text from a script file.

The SCRIPT parameter specifies the file name of the script. A complete file name must be specified, including device, filename and type. The type must be "SCP" or "CFG".

The LINE parameter specifies the line in the script to be deleted. If the LINE parameter is not specified, the entire script is deleted.



For security reasons this command will only be accepted if the user has SECURITY OFFICER privilege.

Examples To delete a script called SHOWME.SCP, use the command:

```
DELETE SCRIPT=SHOWME.SCP
```

See Also `ACTIVATE SCRIPT`
`ADD SCRIPT`
`DEACTIVATE SCRIPT`
`DELETE FILE`
`SET SCRIPT`
`SHOW SCRIPT`

IF..THEN..ELSE..ENDIF

Syntax `IF string1 {EQ|NE} string2 THEN commands [ELSE commands]
ENDIF`

where:

- *string1* and *string2* are character strings, 1 to 255 characters in length. Valid characters are any printable character.

Description The IF..THEN..ELSE..ENDIF control structure is used in a script to execute a different set of router commands depending on some condition:

```
IF string1 {EQ|NE} string2 THEN  
    router commands...  
ENDIF
```

```
IF string1 {EQ|NE} string2 THEN
    router commands...
ELSE
    router commands...
ENDIF
```

The EQ and NE logical operators test that *string1* and *string2* are equal or not equal, respectively. Tests are not case sensitive, so the expressions:

```
FLASH EQ FLASH
FLASH EQ flash
```

are equivalent. *string1* and *string2* may be the replaceable parameters %1 to %8, allowing script execution to be controlled by parameters passed to the script.

If the result of the expression is true, then the router commands between the IF..THEN and ELSE or ENDIF statements are executed. Control continues with the next statement after the IF..THEN..ELSE..ENDIF statement. If the result of the expression is false and there is an ELSE clause, then the router commands between the ELSE and ENDIF statements are executed. Control continues with the next statement after the IF..THEN..ELSE..ENDIF statement. If the result of the expression is false and there is no ELSE clause, then control continues with the next statement after the IF..THEN..ELSE..ENDIF statement.

Examples The following script, named L.SCP, illustrates conditional execution based on passed parameters:

```
IF %2 EQ FLASH THEN
    LOAD FILE=%1 DEST=FLASH SERVER=202.36.163.10
ELSE
    LOAD FILE=%1 DEST=FLASH SERVER=202.36.163.10
ENDIF
```

The script could be activated to load the file FILE.TXT into FLASH using the command:

```
ACTIVATE SCRIPT=L.SCP FILE.TXT FLASH
```

See Also WAIT

SET SCRIPT

Syntax SET SCRIPT=*filename* LINE=*line* [AFTER=*line*] [BEFORE=*line*]
[TEXT=*text*]

where:

- *filename* is a file name of the form `device:filename.type`. *device* is the name of the memory device in which the file is stored (e.g. FLASH). *type* must be "SCP" or "CFG". Valid characters are uppercase letters (A–Z), lowercase letters (a–z), digits (0–9) and the hyphen character (-). Wildcards are not allowed.
- *line* is the number of a line in the script, expressed as a decimal number.
- *text* is a character string, 1 to 127 characters in length. Valid characters are any printable character. If the string contains spaces it must be enclosed in double quotes.

Description This command is used to change the contents of a script file, in command line mode.

The SCRIPT parameter specifies the file name of the script. A complete file name must be specified, including device, filename and type. The type must be "SCP" or "CFG".

The LINE parameter specifies the line in the script to be replaced or moved. If the LINE parameter is used with the TEXT parameter, the LINE parameter specifies the line to be replaced and the TEXT parameter specifies the new contents of the line. If the LINE parameter is used with the AFTER or BEFORE parameters, the LINE parameter specifies the line to be moved and the AFTER or BEFORE parameter specifies the new position of the line in the script. One and only one of the parameters AFTER, BEFORE or TEXT must be specified, in addition to the LINE parameter. The parameters AFTER, BEFORE and TEXT are mutually exclusive. A line can be moved or changed, but not moved and changed, in a single command.

The AFTER parameter specifies the new location of the line identified by the LINE parameter in the script file. The line specified by the LINE parameter is moved to the line following the line specified by the AFTER parameter.

The BEFORE parameter specifies the new location of the line identified by the LINE parameter in the script file. The line specified by the LINE parameter is moved to the line immediately preceding the line specified by the BEFORE parameter.

The TEXT parameter specifies the new contents of the line identified by the LINE parameter in the script file. The entire line is replaced.

There are easier methods of changing scripts than using the SET SCRIPT command. Script files can be edited using the built-in editor (see *Chapter 1, Operation*), or edited on another computer system and downloaded to the router using the LOAD command.



For security reasons this command will only be accepted if the user has SECURITY OFFICER privilege.

Examples To change the third line of text in a script called SHOWME.SCP, use the command:

```
SET SCRIPT=SHOWME.SCP LINE=3 TEXT="SHOW TIME"
```

See Also ACTIVATE SCRIPT
ADD SCRIPT
DELETE SCRIPT
DEACTIVATE SCRIPT
SHOW SCRIPT

SHOW SCRIPT

Syntax SHOW SCRIPT [=filename]

where:

- *filename* is a file name of the form `device:filename.type`. *device* is the name of the memory device in which the file is stored (e.g. FLASH). *type* must be "SCP" or "CFG". Valid characters are uppercase letters (A–Z), lowercase letters (a–z), digits (0–9) and the hyphen character (-). Wildcards are not allowed.

Description This command displays the list of scripts stored on the router, or the contents of the specified script file.

The SCRIPT parameter specifies the file name of the script. A complete file name must be specified, including device, filename and type. The type must be "SCP" or "CFG". If a file name is not specified then the list of all scripts stored on the router is displayed (Figure 13-1 on page 13-11, Table 13-1 on page 13-12). If a file name is specified then the contents of that script file are displayed (Figure 13-2 on page 13-12).

Figure 13-1: Example output from the SHOW SCRIPT command.

Configuration Scripts:			
Filename	Device	Size	Created

boot.cfg	flash	127	18-May-1996 11:08:10

General Scripts:			
Filename	Device	Size	Created

acctstd.scp	flash	201	30-May-1996 15:15:39
syn-trig.scp	flash	1910	30-May-1996 14:41:16
syn0-a1.scp	flash	456	22-May-1996 14:11:03
test.scp	flash	13	05-Jun-1996 12:18:56

Figure 13-2: Example output from the SHOW SCRIPT command for a specified script.

```
File : flash:acctstd.scp

1:purge acc
2:set po=2 sp=115.2k cd=connect flow=hard
3:add acc call=dialin dir=ans auth=none encap=none po=2
4:add acc script=reset.mds text="[ate0q1s0=1^M]"
5:set acc call=dialin rscr=reset.mds
```

Table 13-1: Parameters displayed in the output of the SHOW SCRIPT command.

Parameter	Meaning
Filename	The file name of the script file.
Device	The memory device on the router in which the script file is stored; "flash".
Size	The size of the script file, in bytes.
Created	The date and time the script file was created.

Examples To display the list of scripts stored on the router, use the command:

```
SHOW SCRIPT
```

See Also ACTIVATE SCRIPT
ADD SCRIPT
DEACTIVATE SCRIPT
DELETE SCRIPT
SET SCRIPT

WAIT

Syntax WAIT *delay*

where:

- *delay* is a time delay, in seconds.

Description This command pauses execution of the active script for the specified period of time. The WAIT command is only valid when executed from a script, and can not be executed directly from the command line.

Examples To pause the active script for five seconds, use the command:

```
WAIT 5
```

See Also IF..THEN..ELSE..ENDIF

Chapter 14

Telephony Services

Introduction	14-2
Ports	14-2
Extensions	14-3
Groups	14-4
Numbers	14-4
Tones	14-4
Calls	14-6
Call Handling	14-7
Call Waiting	14-8
Conference Calling	14-8
Call Transfer	14-9
Call Forwarding	14-9
Call Processing	14-9
MSN and DDI Support	14-9
B Channel Allocation	14-10
Bearer Capability, LLC and HLC	14-11
Enbloc or Overlap Dialling	14-12
Tone Suppression	14-12
Hardware Configuration	14-12
Call Logging	14-12
Command Reference	14-13
CREATE PBX EXTENSION	14-13
CREATE PBX GROUP	14-16
DESTROY PBX EXTENSION	14-17
DESTROY PBX GROUP	14-18
DISABLE PBX DEBUG	14-18
ENABLE PBX DEBUG	14-19
SET PBX	14-20
SET PBX EXTENSION	14-21
SET PBX GROUP	14-24
SHOW PBX	14-25
SHOW PBX CALL	14-26
SHOW PBX EXTENSION	14-27
SHOW PBX GROUP	14-29

Introduction

This chapter describes the telephony services available on the router, support for voice over ISDN on the router, and how to configure the router to provide telephony services.

Telephony services are only available on routers fitted with both analogue voice (VOX) ports and ISDN Basic Rate (BRI) interfaces. Each VOX port allows a single standard DTMF analogue telephone to be connected to the router to make calls via an ISDN connection.

The PBX module interfaces to the VOX hardware to detect hardware events (e.g. off hook and DTMF key press), control hardware functions (e.g. tone generation and ringing), and provide support for voice over ISDN.

When the router boots, the PBX module is automatically configured to preset default values. Each port is assigned a default extension number corresponding to the port number. For example, port 0 is assigned extension number '0'. A default group containing all extensions is created to answer all incoming calls and to ring all extensions in the group at once. The default settings allow immediate access to the PBX voice functionality without requiring any further configuration. The boot script can be used to implement a custom setup.

The PBX module provides extended PABX functionality, with features and capabilities normally only found on larger dedicated PABX systems, including:

- **External Calls:** Attempt or accept external phone calls via a local phone.
- **Call Logging:** Record information about calls made.
- **Reserve B Channels:** Reserve one B channel for voice calls and one B channel for data calls.
- **Multiple Subscriber Numbering (MSN) Support:** Process incoming calls depending on the received MSN digit.
- **Direct Dial In (DDI) Support:** Process incoming calls depending on the dialled number.
- **Call Priority:** Requisition an engaged B channel for a higher priority call.
- **Group Hunting:** Specify a group of phones to answer an incoming call, at once or in turn.
- **Tone Suppression:** Delay the activation of tone after a call has ended.
- **Enbloc or Overlap Dialling:** Specify Enbloc dialling (all digits are sent in a single call setup message), or overlap dialling (digits are passed individually to the network).

Ports

The availability of telephony services depends on the presence or absence of VOX ports. If one or more VOX ports are detected when the router boots, the PBX module is initialised. Each VOX port is represented in the PBX module by a PBX port number. The ports are numbered from 0 to $n-1$ for a router with n VOX ports.

Extensions

Individual extension numbers are associated with ports. Any valid extension number may be associated with a port, and does not have to reflect the physical VOX port number. Extension numbers provide a reference to information about the user of the phone attached to the VOX port.

All information about the user of a phone, such as the user's name and active features, is associated with the extension number of the phone.

By associating a user with an extension number, the user's personal information is abstracted from the physical port location. When a user transfers to a different physical port, the user's current extension number can simply be moved to the new port and all the information about the user moves with the extension number.

Extension numbers may be 1 to 3 characters in length. All extensions must be unique, and this applies on a leading digit basis. For example, if one extension number is set to the value '1' then longer extension numbers beginning with '1' can not be used, so an extension of '10' could not subsequently be defined. Extension numbers of identical length may have the same leading digits, so extensions of '10' and '11' may coexist.

Extensions can be configured to accept specific types of incoming calls, depending on call information. For example, incoming calls can be passed to an extension depending on the received MSN or DDI number.

When the router boots, the PBX module is automatically configured to preset default values. Each port is assigned a default extension number corresponding to the port number. For example, port 0 is assigned extension number '0'. The default settings allow immediate access to the PBX voice functionality without requiring any further configuration. The boot script can be used to implement a custom setup.

To create an extension or modify an existing extension, use the commands:

```
CREATE PBX EXTENSION=extension-number PORT={pbx-interface|
NONE} [BCAP={SPEECH|AUDIO}] [COPY=extension-number]
[GROUP=group-name] [HLC={DEFAULT|FAX|TELEPHONE}]
[NAME=extension-name] [NOHLC={ACCEPT|REJECT}]
[NUMACCEPT={matching-number|ALL|NOTPRESENT|OFF}]
[SUBACCEPT={matching-subaddr|ALL|NOTPRESENT|OFF}]
[SUPPRESS={1..30|NONE}] [TERMINATE={1..30|NONE}]

SET PBX EXTENSION=extension-number [BCAP={SPEECH|AUDIO}]
[COPY=extension-number] [GROUP=group-name] [HLC={DEFAULT|
FAX|TELEPHONE}] [NAME=extension-name] [NOHLC={ACCEPT|
REJECT}] [NUMACCEPT={matching-number|ALL|NOTPRESENT|OFF}]
[PORT={pbx-interface|NONE}] [SUBACCEPT={matching-subaddr|
ALL|NOTPRESENT|OFF}] [SUPPRESS={1..30|NONE}]
[TERMINATE={0..30|NONE}]
```

To destroy an existing extension, use the command:

```
DESTROY PBX EXTENSION=extension-number
```

Groups

Groups are lists of extensions that are used to group users together and store specific information about them. Groups are an efficient way of applying the same call acceptance, call barring and call override settings to several extensions.

Groups can be set to accept specific types of incoming calls, based on MSN number or DDI number. When an incoming call is received for a group the hunting facility may be set to either ring all extensions in the group at once (no hunting), or to ring each extension in turn, diverting after a set time (hunting). The order in which the extensions are entered in the list determines the order in which they are hunted. If an extension within the hunt group has a divert set, the call will be diverted to the specified extension, but will fall back into the group hunt if it is not answered.

To create a group or modify an existing group, use the commands:

```
CREATE PBX GROUP=group-name [EXTENSION=extension-number]
    [HUNT={ SEARCH|NONE}] [NUMACCEPT={matching-number|ALL|
    NOTPRESENT|OFF}] [SUBACCEPT={matching-subaddr|ALL|
    NOTPRESENT|OFF}]
SET PBX GROUP=group-name [EXTENSION=extension-number]
    [HUNT={ SEARCH|NONE}] [NUMACCEPT={matching-number|ALL|
    NOTPRESENT|OFF}] [SUBACCEPT={matching-subaddr|ALL|
    NOTPRESENT|OFF}]
```

To destroy an existing extension, use the command:

```
DESTROY PBX GROUP=group-name
```

On initialisation a default group containing all extensions is created to answer all incoming calls and to ring all extensions in the group at once.

Numbers

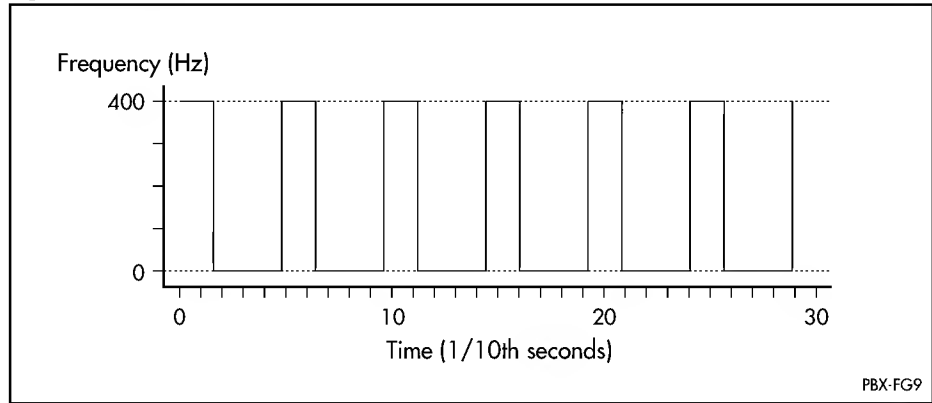
Numbers may be from 1 to 32 numeric characters in length. When a number is dialled from an extension, each digit is checked as it is received. If the PBX module determines that an invalid number has been dialled, the user is informed of this by the tone heard on the receiver.

Tones

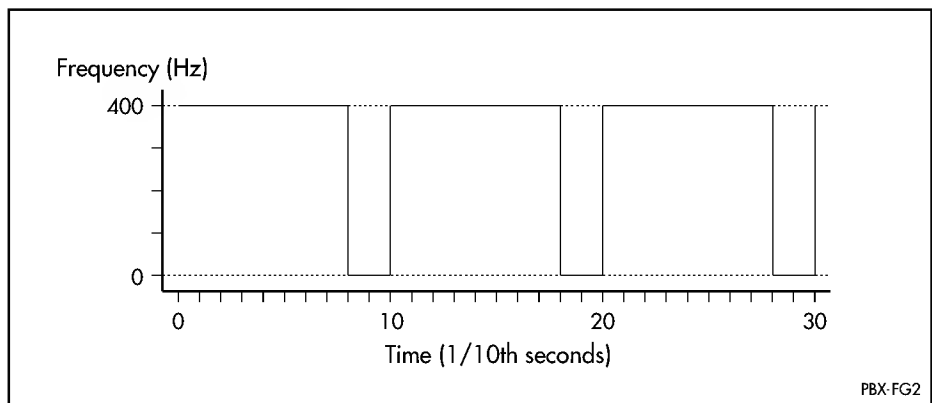
Tones are used to inform the user of the progress of the call or operation. Tones are generated in the earpiece of the connected phone and are distinguished by the cadencing, or on-off time periods, of a 400Hz tone. Each tone indicates a different operational state. For example, if a user off-hooks a phone they will hear the dial tone; if the user dials a valid extension, they will hear the ringing tone. The default tone cadences depend on the PBX 'country'. The tones shown in the following graphs are for the PBX 'country' set to UK.

Bell Tone

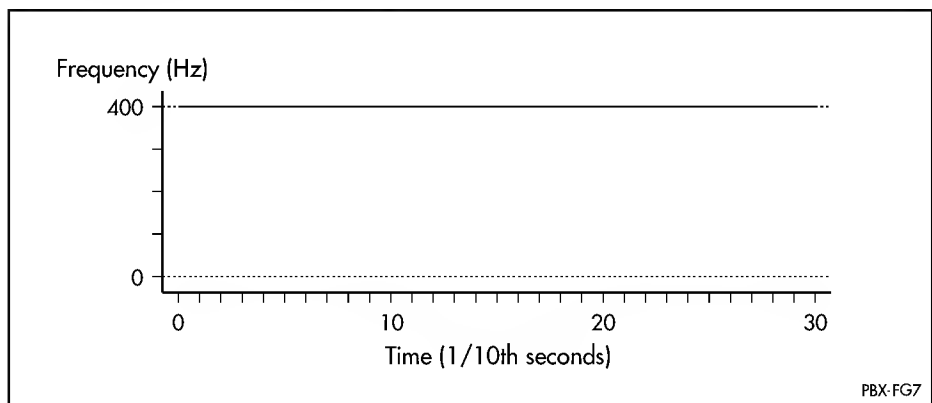
The bell tone (Figure 14-1 on page 14-5) indicates an incoming. This tone is applied when the call setup message is received from the network, and is cleared when the phone is off-hooked.

Figure 14-1: Bell Tone.**Unavailable Tone**

The unavailable tone (Figure 14-2 on page 14-5) indicates that the operation selected is currently unavailable, or that the requested feature is illegal or can not be performed. This tone is applied after the user has selected an illegal or unavailable operation. The phone must be on-hooked to clear the tone.

Figure 14-2: Unavailable Tone.**External Dial Tone**

The external dial tone (Figure 14-3 on page 14-5) indicates that an external number may be dialed. It is applied when enbloc dialling is selected. The tone is cleared when the first digit of the external number is entered.

Figure 14-3: External Dial Tone.

The cadencing for each tone can be modified using the command:

```
SET PBX CADENCE={BELL|UNAV} VALUE=on1,off1,on2,off2,on3,off3
```

Calls

Calls involve the connection of an internal phone and an external phone on the PSTN to allow voice communication.

External calls are calls made to, or accepted from, the external telephone network. The number of external calls that can be made depends on the number of available B channels.

Voice calls must compete with data calls to gain access to available B channels. To manage this conflict, the priority of data calls can be set so that they override, or are overridden by voice calls, as required (Table 14-1 on page 14-6).

Table 14-1: Call priority and call bumping.

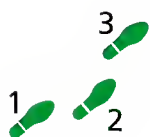
Data calls of priority...	Are bumped by...
0-19	Incoming or outgoing voice calls and incoming or outgoing data calls of higher priority.
20-39	Outgoing voice calls and incoming or outgoing data calls of higher priority.
40-59 (including the default priority of 50)	High priority outgoing voice calls and incoming or outgoing data calls of higher priority.
60-99	High priority outgoing voice calls and outgoing data calls of higher priority.



Although Table 14-1 specifies that a call can be bumped by an incoming call, it is likely that the ISDN to which the router is attached will not offer another incoming call if all B channels are in use. Instead a busy signal will be returned to the originating caller.

If there are no free B channels available then external calls can not be made or accepted. The user wanting to make an external call must wait until a B channel becomes free. This situation is called *external blocking*. If the B channel is occupied by a data call with a low priority then it will be *bumped* (disconnected) in favour of an outgoing voice call.

If more than one incoming call is waiting to be answered and ringing the same group, the first extension to off-hook accepts the first incoming call, and the second extension to off-hook accepts the second incoming call. This feature is called call queuing.



To make an external call:

1. Off-hook the phone.

The external network's dial tone (for overlap dialling) or the external dial tone (for enbloc dialling) will be heard.

If there are no free B channels on which to make the call, the unavailable tone is heard. The phone must be on-hooked to clear the tone. The call may be re-attempted when a B channel becomes available.

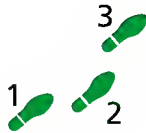
2. Dial the required telephone number.

The network mute tone will be heard after the first digit.

If an invalid number is dialled, the network's unavailable tone is heard. The phone must be on-hooked to clear the tone.

3. The external network's ring tone will be heard.

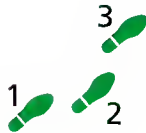
If the dialled number is off-hook or engaged, the network's busy tone will be heard. The phone must be on-hooked to clear the tone.

4. The remote dialled phone will ring.**To accept an incoming external call:****1. The extensions that can receive the call are ringing.**

When an incoming external call is received, the call is directed to one or more ports based on the call acceptance settings of extensions and groups. By default all extensions will ring at once for an incoming call.

2. Off-hook any of the ringing phones to answer the call.

The answering extension is connected to the incoming call. All other ringing phones for that call stop ringing.

**To clear an external call:****1. On-hook the phone.**

The other user will hear the network's unavailable tone. The phone must be on-hooked to clear the tone.

Call Handling

The router supports ISDN supplemental services. These services must be provisioned on your ISDN line before they can be used by the router. The supplementary services supported are:

- Call Waiting
- Conference (3-way) Calling
- Implicit Call Transfer
- Call Forwarding
- Reminder Ring

Call Waiting requires the *Additional Call Offering* (ACO) service to be provisioned on the ISDN line. Conference Calling and Call Transfer require the *Flexible Calling* service to be provisioned on the ISDN line. Check with your telephone company to confirm that these services are available and if there are any additional charges for these services.

Typically, supplementary services are only supported on the first directory number. If supplementary services are required on additional directory numbers, this must be explicitly request, and will usually cost more.

Call Waiting

Call waiting allows a call to be placed on hold while another incoming call on the same directory number is answered. By default call waiting is disabled on both extensions. It can be enabled (or disabled) on either extension using the command:

```
SET PBX EXTENSION=extension-number CALLWAITING={ON|OFF}
```

Call waiting on the ISDN line works the same way it does on a regular analog line. A call waiting indicator tone indicates that there is another incoming call.

- To put the current call on hold and answer the incoming call, press and release the flash hook button on the telephone.
- To switch back to the first call, press and release the flash hook button on the telephone.
- To toggle between the two calls, press and release the flash hook button.
- To end the current call when the call waiting tone is heard and answer the second incoming call, hang-up the telephone and wait for the telephone to ring again. Pick up the telephone to answer the second call.
- To end the current call when there is a call on hold, hang-up the telephone and wait for the telephone to ring again. Pick up the telephone to reconnect to the call on hold.
- To end the call on hold, reconnect to the held call by pressing and releasing the flash hook button. Then hang-up the telephone and wait for the telephone to ring again. Pick up the telephone to return to the original call.
- If the caller on hold hangs up, attempting to reconnect to the held call will either put the current call on hold (if Flexible Calling is not available) or dial tone will be heard (if Flexible Calling is available). If the current call is placed on hold, press and release the flash hook button on the telephone to reconnect to the call.
- If the current caller hangs up and there is a call on hold, either press and release the flash hook button on the telephone to reconnect to the held call or hang-up the telephone and wait for the telephone to ring again. Pick up the telephone to reconnect to the call on hold.

Conference Calling

The Conference Call feature allows a third party to be added to an existing call. Flexible Calling must be enabled on the ISDN line. A conference call can be initiated by either the calling party or answering party of the original call.

- To add a third party to the existing call, press and release the flash hook button on the telephone to put the existing call on hold and receive a dial tone. Dial the third party's telephone number. When the third party answers, press and release the flash hook button on the telephone to establish the three-way conference call. The third party can be conferenced into the call before the third party has answered the call. A third party can not be added to a call until the initial call has been answered.
- To drop the last call added to the conference call, press and release the flash hook button on the telephone. The last call that was added to the conference will be dropped.
- To disconnect from the conference call, hang-up the telephone. The other two callers will remain connected and the ISDN network will connect the two remaining callers together.

- If the third party's line is busy, press and release the flash hook button on the telephone once to return to the original caller. If the third party does not answer, and the third party has not already been added to the conference call, press and release the flash hook button on the telephone twice to return to the original caller. If the third party has already been added to the conference call, press and release the flash hook button on the telephone once to return to the original caller.
- If an incorrect or unallocated number is dialled, wait until the unavailable tone is heard. Then press and release the flash hook button on the telephone once to return to the original caller. If a wrong number is dialled, press and release the flash hook button on the telephone once to drop the call and again to return to the original call.

Call Transfer

Call transfer allows an active call to be implicitly transferred to a third party. Flexible Calling must be enabled on the ISDN line.

- To transfer an active call to a third party and announce the call, press and release the flash hook button on the telephone to put the active call on hold and receive a dial tone. Then dial the third party's phone number. Announce the call, then press and release the flash hook button to establish a conference call. Hang-up the telephone. The ISDN network will implicitly transfer to directly connect the two callers.
- To transfer an active call to a third party, without announcing the call (known as a blind transfer), press and release the flash hook button on your telephone to put the active call on hold and receive a dial tone. Then dial the third party's phone number. Before the third party answers the call, hang-up the phone. The ISDN network will connect the two callers.

Call Forwarding

Call forwarding allows the current call to be forwarded to another telephone number. To use call forwarding, dial the appropriate dial access codes on the telephone keypad. Your telephone company will provide you with the access code. In most cases dialling "*72" followed by the new telephone number will activate the function and dialling "*73" will deactivate the forwarding.

While the call forwarding feature is active, the forwarded call is announced by a single ring on the telephone from which the access codes were entered. This feature is just to remind you that your calls are being forwarded.

Call Processing

Call processing determines how calls are treated, and includes the routing of incoming calls, allocation of ISDN B channels and interaction with ISDN switches.

MSN and DDI Support

When an incoming call is received a method of selecting which phone(s) to ring is needed. For *Multiple Subscriber Number* (MSN) capable lines, or for *Direct Dial In* (DDI) capable lines, the same ISDN connection can be reached by different dialled numbers.

The last digits of the number can be used to select how the call is processed. Calls can be routed to different groups or extensions by checking the dialled number. An accept number, up to 32 digits long, can be defined and then used to match against an incoming call's dialled number. The matching takes place from the last dialled digit.

The router looks for the dialled number in the called number information element (IE) of the Q.931 setup message for the incoming call. In some countries the called number IE may not be provided by the network, but there may be a subaddress IE that contains the last digits dialled. Calls can be routed by checking the called number or subaddress IE. A subaddress match takes precedence over a called number match. Calls with a setup message that does not include either of these IEs may also be routed to an extension or group.

A wildcard character can be used as a placeholder for the last digits of the accept, to add flexibility. For example, this would allow a customer service group to accept any incoming calls to numbers ending in 3X, but if an extension has an accept number of 31 and the incoming call's last digits were 31, then the call would go to that extension.

A group can be configured to answer to particular dialled numbers using the commands:

```
CREATE PBX GROUP=group-name NUMACCEPT={matching-number|ALL|
NOTPRESENT|OFF} SUBACCEPT={matching-subaddr|ALL|
NOTPRESENT|OFF}
SET PBX GROUP=group-name NUMACCEPT={matching-number|ALL|
NOTPRESENT|OFF} SUBACCEPT={matching-subaddr|ALL|
NOTPRESENT|OFF}
```

The group's default call handling can be overridden for an individual extension using the commands:

```
CREATE PBX EXTENSION=extension-number PORT={pbx-interface|
NONE} NUMACCEPT={matching-number|ALL|NOTPRESENT|OFF}
SUBACCEPT={matching-subaddr|ALL|NOTPRESENT|OFF}
SET PBX EXTENSION=extension-number
NUMACCEPT={matching-number|ALL|NOTPRESENT|OFF}
SUBACCEPT={matching-subaddr|ALL|NOTPRESENT|OFF}
```

B Channel Allocation

Data call priorities may be set to different values to give control over channel allocation to ensure that there will always be a channel available for voice calls. See *"Call Control"* on page 4-22 of *Chapter 4, Integrated Services Digital Network (ISDN)* for more information about data call priorities and call bumping.

One B channel can be reserved for voice traffic and the other B channel reserved for data traffic using the command:

```
SET PBX RESERVEBCHANNEL={ON|OFF}
```

B channel reservation ensures that there is always a channel clear and prevents voice calls from occupying both channels. For example, in a retail site one channel may be reserved for credit card authorisations. When a call is attempted and a call of that type is already in progress, the second call is rejected.

Bearer Capability, LLC and HLC

When the router makes an ISDN call for the PBX module it can specify the values in three of the information elements (IEs) in the SETUP message—Bearer Capability, Low Layer Compatibility (LLC) and High Layer Compatibility (HLC). The values are controlled by the BCAP and HLC parameters in the commands:

```
CREATE PBX EXTENSION=extension-number [BCAP={SPEECH|AUDIO}]
[HLC={DEFAULT|TELEPHONE|FAX}]
SET PBX EXTENSION=extension-number [BCAP={SPEECH|AUDIO}]
[HLC={DEFAULT|TELEPHONE|FAX}]
```

The bearer capability indicates to the network the quality of circuit that is required. The value SPEECH is appropriate for voice calls and AUDIO (3.1 kHz) for modem or fax machine calls. The bearer capability may be set for each extension. The default is AUDIO. If a modem or fax machine is connected to a VOX port then the bearer capability for the associated extension should be set to AUDIO.

The LLC indicates to the device being called the bearer capability of the call. It is always included in the outgoing SETUP message and will have identical contents to the bearer capability.

The HLC indicates to the device being called the high layer protocol for the call. This IE is not normally included in outgoing SETUP messages, that is, when the extension's HLC parameter is set to DEFAULT. When the HLC parameter is set to FAX or TELEPHONE, the HLC IE is included and set to "telephony" and "group 2/3 fax" respectively.

When both BCAP and HLC parameters are used together, the Bearer Capability and LLC are set to SPEECH, except when either the BCAP or HLC parameters indicate that 3.1 kHz audio should be used instead, that is, when BCAP is set to AUDIO or HLC is set to FAX (Table 14-2 on page 14-11).

Table 14-2: Affect of BCAP and HLC parameters on ISDN call SETUP messages.

Command Parameters		Resulting SETUP message contents		
BCAP	HLC	Bearer Capability IE	LLC (Low Layer Compatibility) IE	HLC (High Layer Compatibility) IE
(default)	(default)	SPEECH	SPEECH	(not present)
(default)	TELEPHONE	SPEECH	SPEECH	telephony
(default)	FAX	AUDIO	AUDIO	group 2/3 fax
SPEECH	(default)	SPEECH	SPEECH	(not present)
SPEECH	TELEPHONE	SPEECH	SPEECH	telephony
SPEECH	FAX	AUDIO	AUDIO	group 2/3 fax
AUDIO	(default)	AUDIO	AUDIO	(not present)
AUDIO	TELEPHONE	AUDIO	AUDIO	telephony
AUDIO	FAX	AUDIO	AUDIO	group 2/3 fax

Enbloc or Overlap Dialling

The normal method of making a ISDN phone call is to send each digit to the network as it dialled. This is called *overlap dialling*. Some networks require all the dialled digits on the ISDN network side to arrive within one Q.931 SETUP message. This is called *enbloc dialling*. Enbloc or overlap dialling can be configured on a global basis, using the command:

```
SET PBX DIAL={OVERLAP|ENBLOC}
```

When enbloc dialling is selected the dialled telephone number must be terminated with the "data" key ("*" or "#" key) so that the router knows when all digits have been dialled and the Q.931 SETUP message may be sent. The data key defaults to "#" and may be changed to "*", if desired, with the SET PBX command on page 14-20. In some installations the requirement that the number be terminated with the data key may be highly inconvenient, for example when a fax machine that has already been programmed with many numbers (not terminated with a "#" or "*") is attached to a router VOX port. In this case the router may be programmed to terminate the number automatically after a specified time interval following the last received digit. This feature is normally only useful in the situation where a fax machine is attached since it dials the digits quickly and a pause of just a second or two is enough to indicate that all the digits have been dialled.

Tone Suppression

Answering machines listen for a period of silence to detect that the call has hung up and then stop recording. To provide for this the internal tone must be suppressed for a configurable period. Tone suppression can be configured for individual extensions, using the command:

```
SET PBX EXTENSION=extension-number SUPPRESS={1..30|NONE}
```

Hardware Configuration

The VOX hardware configuration may be customised to suit local or national standards or common practises.

The type of speech encoding used on the B channels can be configured for compatibility with different networks. For the U.S. and Japan uLaw encoding is used, whereas in Europe and other countries aLaw encoding is used.

Tone cadencing can be configured to give custom values for the on-off periods of each tone. The bell cadence can also be changed. Profiles for specific countries can be selected to simplify operation.

Call Logging

Information about the calls currently in progress can be displayed using the command:

```
SHOW PBX CALL
```

The router's logging facility is used to provide a history of past completed calls. For incoming calls the log includes the group to which the call was directed if the call was not answered or the extension that answered the call,

the remote calling number (if available), who terminated the call and the call duration. For outgoing calls the log includes the extension from which the call was made, the called number, who terminated the call and the call direction. The call log can be displayed using the command:

```
SHOW LOG
```

Command Reference

This section describes the commands available on the router to configure and manage the telephony services. Telephony services require ISDN to be enabled and configured correctly. See *Chapter 4, Integrated Services Digital Network (ISDN)* for detailed descriptions of the commands required to enable and configure ISDN.

See “Conventions” on page xxxv of *Preface* in the front of this manual for details of the conventions used to describe command syntax. See *Appendix A, Messages* for a complete list of messages and their meanings.

CREATE PBX EXTENSION

Syntax `CREATE PBX EXTENSION=extension-number [BCAP={SPEECH|AUDIO}] [CALLINGNUMBER={calling-number|OFF}] [COPY=extension-number] [HLC={DEFAULT|FAX|TELEPHONE}] [GROUP=group-name] [NAME=extension-name] [NOHLC={ACCEPT|REJECT}] [NUMACCEPT={matching-number|ALL|NOTPRESENT|OFF}] [PORT={pbx-interface|NONE}] [SUBACCEPT={matching-subaddr|ALL|NOTPRESENT|OFF}] [SUPPRESS={1..30|NONE}] [TERMINATE={0..30|NONE}]`

where:

- *extension-number* is a string of decimal digits (0–9), 1 to 3 characters in length.
- *calling-number* is a string of decimal digits (0–9), 1 to 7 characters in length.
- *group-name* is a string, 1 to 15 characters in length. Valid characters are uppercase letters (A–Z), lowercase letters (a–z), digits (0–9) and spaces. If *group-name* contains spaces it must be enclosed in double quotes.
- *extension-name* is a string, 1 to 15 characters in length. Valid characters are uppercase letters (A–Z), lowercase letters (a–z) and digits (0–9).
- *matching-number* is a string, 1 to 6 characters in length. Valid characters are decimal digits (0–9) and the wildcard characters “x” and “X”.
- *pbx-interface* is a decimal number, ranging from 0 to the number of PBX ports on the router minus 1.
- *matching-subaddr* is a string, 1 to 6 characters in length. Valid characters are decimal digits (0–9), the wildcard characters “x” and “X”, and the keypad character “*”.

Description This command creates an extension in the PBX module, associates it with a particular PBX port and sets other operational parameters for the extension.

The EXTENSION parameter specifies the extension to create. The new extension must not match any extension already on the router.

The BCAP parameter specifies the bearer capability required for calls made from this extension. If SPEECH is specified, the router will request a speech-grade circuit from the ISDN. If AUDIO is specified, the router will request a 3.1 kHz audio-grade circuit from the ISDN, which is appropriate for an extension to which a modem or fax machine is connected. The default is SPEECH. Note however, that if the HLC parameter is set to FAX, the setting of BCAP to SPEECH will be overridden and the bearer capability and low layer compatibility will be set to 3.1 KHz audio.

The CALLINGNUMBER parameter specifies the format of the calling party number IE (also known as CLI) in the outgoing call SETUP message created when calls are made from this extension. The STD number is not included. If OFF is specified, the calling party number IE is not included in the call SETUP message. The default is OFF.

The COPY parameter allows the newly created extension to be based on an existing extension's configuration. The value of the COPY parameter is the extension number of an existing extension, whose configuration is copied into the configuration of this extension. Any other configuration parameters that appear in the command will override the values obtained from the copied extension. The PORT parameter must also be specified.

The HLC parameter specifies the contents of the Bearer Capability IE, Low Layer Compatibility (LLC) IE and High Layer Compatibility (HLC) IE in call SETUP messages for calls made from this extension. If DEFAULT is specified, the Bearer Capability and Low Layer Compatibility IEs will be set to speech-grade, and there will be no High Layer Compatibility IE. If TELEPHONE is specified, the Bearer Capability and Low Layer Compatibility IEs will be set to speech-grade and the High Layer Compatibility IE will be set to "telephony". If FAX is specified, the Bearer Capability and Low Layer Compatibility IEs will be set to 3.1 kHz audio-grade and the High Layer Compatibility IE will be set to "group 2/3 fax". Note however, that if the BCAP parameter is set to AUDIO, the setting of HLC to DEFAULT or TELEPHONE will be overridden and the Bearer Capability and Low Layer Compatibility IEs will be set to 3.1 KHz audio. The default is DEFAULT.

The GROUP parameter specifies the PBX extension group to which this extension will belong. The extension group must already exist on the router. If the parameter is not present, the extension will be added to the default group.

The NAME parameter specifies a descriptive name to identify the extension. This could be the physical location of the extension or the name of the person who uses that extension.

The NOHLC parameter specifies whether to accept or reject incoming calls which do not have the High Layer Compatibility IE present in the SETUP message. The default is ACCEPT. Note that a call may be rejected for other reasons even if there is no HLC information and the NOHLC parameter is set to ACCEPT.

The NUMACCEPT parameter specifies the number to match against the called number information element (IE) of an incoming call setup message to determine whether or not to accept the call. If the called number IE in an incoming call matches the value, the call is accepted. The value may contain the wildcard characters "x" and "X" which match any single character. The comparison starts with the last digit of the called number, and proceeds back

towards the first digit of the called number, until a match is found. For example, "x" will match any called number of any length, "3x" will match numbers ending in a 3 followed by any digit, and "9" will match any number ending in 9. The value ALL will match any call. The value NOTPRESENT will match a call with a call setup message that does not include the called number IE. The value OFF will prevent a match regardless of the presence or contents of the called number IE. Subaddress matches take precedence over called number matches, extension matches take precedence over group matches and exact matches take precedence over wildcard matches. The default is NONE.

The PORT parameter specifies the PBX port that will be associated with this extension. The port must not already be associated with any other extension. The value NONE can be specified to create an extension not specifically connected to a PBX port.

The SUBACCEPT parameter specifies the number to match against the subaddress information element (IE) of an incoming call setup message to determine whether or not to accept the call. If the subaddress IE in an incoming call matches the value, the call is accepted. The value may contain the wildcard characters "x" and "X" which match any single character. The comparison starts with the last digit of the subaddress, and proceeds back towards the first digit of the subaddress, until a match is found. For example, "x" will match any subaddress of any length, "3x" will match numbers ending in a 3 followed by any digit, and "9" will match any number ending in 9. The value ALL will match any call. The value NOTPRESENT will match a call with a call setup message that does not include the subaddress IE. The value OFF will prevent a match regardless of the presence or contents of the subaddress IE. Subaddress matches take precedence over called number matches, extension matches take precedence over group matches and exact matches take precedence over wildcard matches. The default is NONE.

The SUPPRESS parameter specifies the timer value, in seconds, for delaying the application of unavailable tone after the remote end of a connected call hangs up. When the remote end of a connected call hangs up, an unavailable tone will be supplied to the local end after the suppress time period. The suppress timer can be turned off by specifying the value NONE. The default is NONE.

The TERMINATE parameter enables the auto-termination feature for enbloc dialling. The value is the duration, in seconds, of a timer that is started each time a digit is dialled after the external prefix. If the timer expires before another digit is dialled, and the timer restarted, then the router assumes that all the digits of the external number have been received and causes Q.931 to send a SETUP message. The feature is disabled by setting the value to NONE. The default is NONE.

Examples To create a new extension 1 on PBX port 1, based on extension 0 but with a different name, use the command:

```
CREATE PBX EXTENSION=1 PORT=1 COPY=0 NAME="GroundFloorLab"
```

To create extension 0 with the calling number "3890123", use the command:

```
CREATE PBX EXTEN=0 PORT=0 CALLINGNUMBER=3890123
```

See Also SET PBX EXTENSION
DESTROY PBX EXTENSION
SHOW PBX EXTENSION

CREATE PBX GROUP

Syntax CREATE PBX GROUP=*group-name* [EXTENSION=*extension-number*]
 [HUNT={SEARCH|NONE}] [NUMACCEPT={*matching-number*|ALL|
 NOTPRESENT|OFF}] [SUBACCEPT={*matching-subaddr*|ALL|
 NOTPRESENT|OFF}]

where:

- *group-name* is a string, 1 to 15 characters in length. Valid characters are uppercase letters (A–Z), lowercase letters (a–z), digits (0–9) and spaces. If *group-name* contains spaces it must be enclosed in double quotes.
- *extension-number* is a string of decimal digits (0–9), 1 to 3 characters in length.
- *matching-number* is a string, 1 to 6 characters in length. Valid characters are decimal digits (0–9) and the wildcard characters “x” and “X”.
- *matching-subaddr* is a string, 1 to 6 characters in length. Valid characters are decimal digits (0–9), the wildcard characters “x” and “X”, and the keypad character “*”.

Description This command creates an extension group in the PBX module. Groups are used to collect extensions together to apply similar features to multiple extensions. For example, a “Customer Service” group could be created so that when a customer call was received, all the extensions within the group would ring.

The GROUP parameter specifies the name of the group to create. The group name must be different from the name of any group already defined in the router. Comparisons for uniqueness are done without regard to the case of letters. For example “TEST” and “test” are equivalent.

The EXTENSION parameter specifies the group’s reference extension number. The extension number must be different from all other group reference extensions already defined in the router.

The HUNT parameter specifies the type of hunting in use. This determines how the group’s phones are rung when a call is received for the group. A value of SEARCH causes one phone to ring at a time, and a divert to the next phone occurs if the extension is engaged or does not answer. A value of NONE causes all phones within the group to ring at once when an incoming call arrives.

The NUMACCEPT parameter specifies the number to match against the called number information element (IE) of an incoming call setup message to determine whether or not to accept the call. If the called number IE in an incoming call matches the value, the call is accepted. The value may contain the wildcard characters “x” and “X” which match any single character. The comparison starts with the last digit of the called number, and proceeds back towards the first digit of the called number, until a match is found. For example, “x” will match any called number of any length, “3x” will match numbers ending in a 3 followed by any digit, and “9” will match any number ending in 9. The value ALL will match any call. The value NOTPRESENT will match a call with a call setup message that does not include the called number IE. The value OFF will prevent a match regardless of the presence or contents of the called number IE. Subaddress matches take precedence over called number matches, extension matches take precedence over group matches and exact matches take precedence over wildcard matches. The default is NONE.

The SUBACCEPT parameter specifies the number to match against the subaddress information element (IE) of an incoming call setup message to determine whether or not to accept the call. If the subaddress IE in an incoming call matches the value, the call is accepted. The value may contain the wildcard characters "x" and "X" which match any single character. The comparison starts with the last digit of the subaddress, and proceeds back towards the first digit of the subaddress, until a match is found. For example, "x" will match any subaddress of any length, "3x" will match numbers ending in a 3 followed by any digit, and "9" will match any number ending in 9. The value ALL will match any call. The value NOTPRESENT will match a call with a call setup message that does not include the subaddress IE. The value OFF will prevent a match regardless of the presence or contents of the subaddress IE. Subaddress matches take precedence over called number matches, extension matches take precedence over group matches and exact matches take precedence over wildcard matches. The default is NONE.

Examples To create a group called "Sales" that accepts all calls and rings extensions in turn until one answers, use the command:

```
CREATE PBX GROUP="Sales" ACCEPT=ALL HUNT=SEARCH
```

See Also DESTROY PBX GROUP

DESTROY PBX EXTENSION

Syntax DESTROY PBX EXTENSION=*extension-number*

where:

- *extension-number* is a string of decimal digits (0–9), 1 to 3 characters in length.

Description This command destroys an extension in the PBX module. When destroying an extension any barred or override numbers associated with the extension are also deleted.

Any incoming calls that would have been accepted by the extension will now be rejected, unless other groups or extensions are configured to accept them.

The EXTENSION parameter specifies the extension to be destroyed. This extension must exist. Extensions created by the system with the type set to DEFAULT can not be deleted. They are provided so that the PBX module will work with no configuration.

Examples To destroy extension 3, use the command:

```
DESTROY PBX EXTENSION=3
```

See Also CREATE PBX EXTENSION

DESTROY PBX GROUP

Syntax DESTROY PBX GROUP=*group-name*

where:

- *group-name* is a string, 1 to 15 characters in length. Valid characters are uppercase letters (A–Z), lowercase letters (a–z), digits (0–9) and spaces. If *group-name* contains spaces it must be enclosed in double quotes.

Description This command destroys an extension group in the PBX module. When destroying a group any barred or override numbers associated with the group are also deleted. The extensions within the group are not deleted, but they are no longer associated with this group and revert to the default group.

Any incoming calls that would have been accepted by the group will now be rejected, unless other groups or extensions are configured to accept them.

The GROUP parameter specifies the group to be destroyed. The group must already exist in the router.

Examples To destroy the group "Sales", use the command:

```
DESTROY PBX GROUP="Sales"
```

See Also CREATE PBX GROUP

DISABLE PBX DEBUG

Syntax DISABLE PBX DEBUG={ALL|CODEC|COMMAND|COUNTERS|CLID|EVENT|REDIRECTEDNUMBER|TRACE}

Description This command disables the specified PBX debugging option or all PBX debugging. Multiple debugging options can be disabled using successive invocations of the command.

The DEBUG parameter specifies which debugging option is to be disabled. The options allowed and the debugging that results from specifying the option are shown in Table 14-3 on page 14-19. If debugging was enabled using the ENABLE PBX DEBUG=ALL command, then disabling each individual option will not disable all debugging. General PBX state debugging information will still be displayed. To completely disable all PBX debugging, use the DISABLE PBX DEBUG=ALL command.

Examples To enable debugging of PBX state machine events, use the command:

```
DISABLE PBX DEBUG=EVENT
```

See Also ENABLE PBX DEBUG

ENABLE PBX DEBUG

Syntax `ENABLE PBX DEBUG={ALL|CODEC|COMMAND|COUNTERS|CLID|EVENT|REDIRECTEDNUMBER|TRACE} [PORT=port-number]`

where:

- *port-number* is the number of an asynchronous port on the router. Ports are numbered starting at zero (0).

Description This command enables PBX debugging. Debugging information is sent to the port or telnet session from which the command was entered if the PORT parameter was not specified, otherwise it is sent to the specified port. Multiple debugging options can be disabled using successive invocations of the command.

The DEBUG parameter specifies which debugging option is to be enabled. The options allowed and the debugging that results from specifying the option are shown in Table 14-3 on page 14-19.

Table 14-3: PBX debugging options.

Option	Description
ALL	All debug options.
CODEC	Data read from and written to the CODEC.
COMMAND	POTS driver commands.
COUNTERS	Counters for PBX activity.
CLID	Call Line ID information.
EVENT	PBX state machine events
REDIRECTEDNUMBER	Redirected numbers.
TRACE	Traces function calls within PBX.

The PORT parameter specifies the asynchronous port to which the debug output is to be sent. This enables debugging to be enabled in a script. The default is to send the output to the terminal or Telnet session from which the command was executed. Each time the ENABLE PBX DEBUG command is entered the destination for debugging output is determined using this rule.

Examples To enable debugging of PBX state machine events, use the command:

```
ENABLE PBX DEBUG=EVENT
```

See Also DISABLE PBX DEBUG

SET PBX

Syntax SET PBX [CADENCE={BELL|UNAV} VALUE=*cadence-values*]
 [COUNTRY={AUSTRALIA|CHINA|CUSTOM|HOLLAND|JAPAN|KOREA|
 NEWZEALAND|UK|USA}] [DATA=*data-key*] [DEBUG={ON|OFF}]
 [DIAL={OVERLAP|ENBLOC}] [DISCONNECT=5..10]
 [ENCODE={ULAW|ALAW}] [FLASHHOOKMIN={2..4|OFF}]
 [INTERDIGIT={1..30|NONE}] [RESERVEBCHANNEL={ON|OFF}]

where:

- *extension-number* is a string of decimal digits (0–9), 1 to 3 characters in length.
- *cadence-values* is a comma separated list of exactly six decimal numbers, each in the range of 0 to 255.
- *data-key* is the '*' or '#' character.

Description This command changes the operational parameters of the PBX module. Options set with this command apply to the overall operation of the PBX module and are not specific to a particular PBX port, group or extension.

This command is used to change the cadencing of the bell and tones. Tone cadences are fixed at a 400Hz signal, but the ON/OFF time periods can be changed. These are specified in tenths of seconds. Three ON/OFF periods are set. To obtain a continuous tone, the ON parameters should be set to a non-zero and the OFF parameters to 0.

The CADENCE parameter specifies which tone cadence to change, as follows:

Keyword	Meaning
BELL	The phone's ringing bell cadence.
UNAV	The unavailable tone cadence.

If CADENCE is specified, the VALUE parameter must also be present. The VALUE parameter specifies the ON/OFF periods for the specified cadence, as a comma-separated list of decimal numbers:

VALUE=*on1, off1, on2, off2, on3, off3*

The COUNTRY parameter selects country-specific settings. This changes the cadencing of the tones to comply with various national standard tones.

The DATA parameter specifies the data terminating character. This key is used to signal that a phone number entry has been completed. The default data terminator is the '#' character.

The DEBUG parameter enables or disables the generation of diagnostic messages for troubleshooting.

The DIAL parameter specifies the ISDN calling method to use. This can be set to either OVERLAP or ENBLOC. The default is OVERLAP.

The DISCONNECT parameter specifies the minimum length of time, in tenths of a second, before a call is disconnected. The default is 10.

The ENCODE parameter specifies the type of encoding used for speech calls. The default is ALAW.

The FLASHHOOKMIN parameter specifies the minimum length of time, in tenths of a second, that the phone's hook switch must be depressed to generate a valid flash hook in order to put a call on hold. If OFF is specified, calls can not be put on hold and are disconnected immediately. The default is 2.

The INTERDIGIT parameter specifies the maximum length of time, in seconds, between digits before an unavailable indication is generated. The default is 10. A value of NONE means that any length of time is allowed between digits.

The RESERVEBCHANNEL parameter specifies whether or not one B channel is to be reserved for voice calls and the other B channel is to be reserved for data calls. If OFF is specified B channels are not reserved and up to two voice calls may be made simultaneously. Data calls will be bumped in favour of voice calls if the priority of the data calls is less than 40. If ON is specified only one external voice call may be made at any one time. The default is OFF.

Examples To set the unavailable tone cadence to a value other than the default, use the command:

```
SET PBX CADENCE=UNAV VAL=10,0,10,0,10,0
```

See Also SHOW PBX

SET PBX EXTENSION

Syntax SET PBX EXTENSION=*extension-number* [BCAP={SPEECH|AUDIO}]
 [CALLINGNUMBER={*calling-number*|OFF}] [COEFFICIENT={TH1 |
 TH2 | TH3 | IM1 | IM2 | FRX | FRR | AX | AR | TG1 | TG2 }
 VALUE=*coefficient-values*] [COPY=*extension-number*]
 [GROUP=*group-name*] [HLC={DEFAULT|FAX|TELEPHONE}]
 [NAME=*extension-name*] [NOHLC={ACCEPT|REJECT}]
 [NUMACCEPT={*matching-number*|ALL|NOTPRESENT|OFF}]
 [PORT={*pbx-interface*|NONE}] [SUBACCEPT={*matching-*
subaddr|ALL|NOTPRESENT|OFF}] [SUPPRESS={1..30|NONE}]
 [TERMINATE={0..30|NONE}]

where:

- *extension-number* is a string of decimal digits (0–9), 1 to 3 characters in length.
- *calling-number* is a string of decimal digits (0–9), 1 to 31 characters in length.
- *coefficient-values* is a comma-separated list of exactly six hexadecimal numbers, each in the range 0 to FF.
- *group-name* is a string, 1 to 15 characters in length. Valid characters are uppercase letters (A–Z), lowercase letters (a–z), digits (0–9) and spaces. If *group-name* contains spaces it must be enclosed in double quotes.
- *extension-name* is a string, 1 to 15 characters in length. Valid characters are uppercase letters (A–Z), lowercase letters (a–z), digits (0–9).
- *matching-number* is a string, 1 to 31 characters in length. Valid characters are decimal digits (0–9) and the wildcard characters “x” and “X”.

- *pbx-interface* is a decimal number, ranging from 0 to the number of PBX ports on the router minus 1.
- *matching-subaddr* is a string, 1 to 6 characters in length. Valid characters are decimal digits (0–9), the wildcard characters “x” and “X”, and the keypad character “*”.

Description This command changes the operational parameters of an extension.

The EXTENSION parameter specifies the extension for which the operational parameters are to be changed. The extension must already exist on the router.

The BCAP parameter specifies the bearer capability required for calls made from this extension. If SPEECH is specified, the router will request a speech-grade circuit from the ISDN. If AUDIO is specified, the router will request a 3.1 kHz audio-grade circuit from the ISDN, which is appropriate for an extension to which a modem or fax machine is connected. The default is AUDIO. Note however, that if the HLC parameter is set to FAX, the setting of BCAP to SPEECH will be overridden and the bearer capability and low layer compatibility will be set to 3.1 KHz audio.

The CALLINGNUMBER parameter specifies the format of the calling party number IE (also known as CLI) in the outgoing call SETUP message created when calls are made from this extension. The STD number is not included. If OFF is specified, the calling party number IE is not included in the call SETUP message. The default is OFF.

The COEFFICIENT parameter specifies a CODEC coefficient to change. If COEFFICIENT is specified, VALUE must also be specified. The coefficients should only be changed by qualified personnel as setting incorrect values could stop the telephones from operating correctly.

The COPY parameter allows the extension to be set to the same configuration as an existing extension. The value of the COPY parameter is the extension number of an existing extension, whose configuration is copied into the configuration of this extension. Any other configuration parameters that appear in the command will override the values obtained from the copied extension. The PORT parameter value for the extension is not copied and remains unchanged.

The GROUP parameter specifies the PBX extension group to which this extension will belong. The extension group must already exist on the router.

The HLC parameter specifies the contents of the Bearer Capability IE, Low Layer Compatibility (LLC) IE and High Layer Compatibility (HLC) IE in call SETUP messages for calls made from this extension. If DEFAULT is specified, the Bearer Capability and Low Layer Compatibility IEs will be set to speech-grade, and there will be no High Layer Compatibility IE. If TELEPHONE is specified, the Bearer Capability and Low Layer Compatibility IEs will be set to speech-grade and the High Layer Compatibility IE will be set to “telephony”. If FAX is specified, the Bearer Capability and Low Layer Compatibility IEs will be set to 3.1 kHz audio-grade and the High Layer Compatibility IE will be set to “group 2/3 fax”. Note however, that if the BCAP parameter is set to AUDIO, the setting of HLC to DEFAULT or TELEPHONE will be overridden and the Bearer Capability and Low Layer Compatibility IEs will be set to 3.1 KHz audio. The default is DEFAULT.

The NAME parameter specifies a descriptive name to identify the extension. This could be the physical location of the extension, or the name of the person who uses that extension.

The NOHLC parameter specifies whether to accept or reject incoming calls which do not have the High Layer Compatibility IE present in the SETUP message. The default is ACCEPT. Note that a call may be rejected for other reasons even if there is no HLC information and the NOHLC parameter is set to ACCEPT.

The NUMACCEPT parameter specifies the number to match against the called number information element (IE) of an incoming call setup message to determine whether or not to accept the call. If the called number IE in an incoming call matches the value, the call is accepted. The value may contain the wildcard characters "x" and "X" which match any single character. The comparison starts with the last digit of the called number, and proceeds back towards the first digit of the called number, until a match is found. For example, "x" will match any called number of any length, "3x" will match numbers ending in a 3 followed by any digit, and "9" will match any number ending in 9. The value ALL will match any call. The value NOTPRESENT will match a call with a call setup message that does not include the called number IE. The value OFF will prevent a match regardless of the presence or contents of the called number IE. Subaddress matches take precedence over called number matches, extension matches take precedence over group matches and exact matches take precedence over wildcard matches. The default is OFF.

The PORT parameter specifies the PBX port that will be associated with this extension. The port must not already be associated with any other extension. The value NONE can be specified to disconnect the extension from a port.

The SUBACCEPT parameter specifies the number to match against the subaddress information element (IE) of an incoming call setup message to determine whether or not to accept the call. If the subaddress IE in an incoming call matches the value, the call is accepted. The value may contain the wildcard characters "x" and "X" which match any single character. The comparison starts with the last digit of the subaddress, and proceeds back towards the first digit of the subaddress, until a match is found. For example, "x" will match any subaddress of any length, "3x" will match numbers ending in a 3 followed by any digit, and "9" will match any number ending in 9. The value ALL will match any call. The value NOTPRESENT will match a call with a call setup message that does not include the subaddress IE. The value OFF will prevent a match regardless of the presence or contents of the subaddress IE. Subaddress matches take precedence over called number matches, extension matches take precedence over group matches and exact matches take precedence over wildcard matches. The default is OFF.

The SUPPRESS parameter specifies the timer value, in seconds, for delaying the application of unavailable tone after the remote end of a connected call hangs up. When the remote end of a connected call hangs up, an unavailable tone will be supplied to the local end after the suppress time period. The suppress timer can be turned off by specifying the value NONE.

The TERMINATE parameter enables the auto-termination feature for enbloc dialling. The value is the duration, in seconds, of a timer that is started each time a digit is dialled after the external prefix. If the timer expires before another digit is dialled, and the timer restarted, then the router assumes that all the digits of the external number have been received and causes Q.931 to send a SETUP message. The feature is disabled by setting the value to NONE. The default is NONE.

Examples To set extension 1 to the same configuration as extension 0, use the command:

```
SET PBX EXTENSION=1 COPY=0
```

To set the calling number to 389-0123 for extension 0, use the command:

```
SET PBX EXTEN=0 CALLINGNUMBER=3890123
```

See Also CREATE PBX EXTENSION
DESTROY PBX EXTENSION
SHOW PBX EXTENSION

SET PBX GROUP

Syntax SET PBX GROUP=*group-name* [EXTENSION=*extension-number*]
[HUNT={SEARCH|NONE}] [NUMACCEPT={*matching-number*|ALL|
NOTPRESENT|OFF}] [SUBACCEPT={*matching-subaddr*|ALL|
NOTPRESENT|OFF}]

where:

- *group-name* is a string, 1 to 15 characters in length. Valid characters are uppercase letters (A–Z), lowercase letters (a–z), digits (0–9) and spaces. If *group-name* contains spaces it must be enclosed in double quotes.
- *extension-number* is a string of decimal digits (0–9), 1 to 3 characters in length.
- *matching-number* is a string, 1 to 6 characters in length. Valid characters are decimal digits (0–9) and the wildcard characters “x” and “X”.
- *matching-subaddr* is a string, 1 to 6 characters in length. Valid characters are decimal digits (0–9), the wildcard characters “x” and “X”, and the keypad character “*”.

Description This command changes the operational parameters of an extension group.

The GROUP parameter specifies the name of the group for which the parameters are to be changed. This group must already exist on the router.

The EXTENSION parameter specifies the group’s reference extension number. The extension number must be different from all other group reference extensions already defined in the router.

The HUNT parameter specifies the type of hunting to use. This determines how the group’s phones are rung when a call is received for the group. A value of SEARCH causes one phone to ring at a time, and a divert to the next phone occurs if the extension is engaged or does not answer. A value of NONE causes all phones within the group to ring at once when an incoming call arrives.

The NUMACCEPT parameter specifies the number to match against the called number information element (IE) of an incoming call setup message to determine whether or not to accept the call. If the called number IE in an incoming call matches the value, the call is accepted. The value may contain the wildcard characters “x” and “X” which match any single character. The comparison starts with the last digit of the called number, and proceeds back towards the first digit of the called number, until a match is found. For example, “x” will match any called number of any length, “3x” will match

numbers ending in a 3 followed by any digit, and "9" will match any number ending in 9. The value ALL will match any call. The value NOTPRESENT will match a call with a call setup message that does not include the called number IE. The value OFF will prevent a match regardless of the presence or contents of the called number IE. Subaddress matches take precedence over called number matches, extension matches take precedence over group matches and exact matches take precedence over wildcard matches. The default is OFF.

The SUBACCEPT parameter specifies the number to match against the subaddress information element (IE) of an incoming call setup message to determine whether or not to accept the call. If the subaddress IE in an incoming call matches the value, the call is accepted. The value may contain the wildcard characters "x" and "X" which match any single character. The comparison starts with the last digit of the subaddress, and proceeds back towards the first digit of the subaddress, until a match is found. For example, "x" will match any subaddress of any length, "3x" will match numbers ending in a 3 followed by any digit, and "9" will match any number ending in 9. The value ALL will match any call. The value NOTPRESENT will match a call with a call setup message that does not include the subaddress IE. The value OFF will prevent a match regardless of the presence or contents of the subaddress IE. Subaddress matches take precedence over called number matches, extension matches take precedence over group matches and exact matches take precedence over wildcard matches. The default is OFF.

Examples To change the hunting function for extensions in group "Sales" so that all phones ring at once, use the command:

```
SET PBX GROUP="Sales" HUNT=NONE
```

See Also SHOW PBX GROUP

SHOW PBX

Syntax SHOW PBX

Description This command displays information about general PBX configuration information. (Figure 14-4 on page 14-25, Table 14-4 on page 14-26).

Figure 14-4: Example output from the SHOW PBX command.

```
PBX Module Configuration

General
Country ..... New Zealand
Encode ..... alaw
Dial ..... overlap
Interdigit ..... 10
Data ..... #

Cadence:
Bell ..... 16 32 0 0 0 0
Unavailable ..... 8 2 8 2 8 2
```

Table 14-4: Parameters displayed in the output of the SHOW PBX command.

Parameter	Meaning
General	General PBX configuration settings.
Country	The country used to set the country specific settings, such as tones and default prefix numbers; one of "Japan", "UK", "Holland", "Australia", "New Zealand", "USA", "China", "Korea" or "custom".
Encode	The encoding method used for speech calls; one of "ulaw" or "alaw".
Dial	The dial method used for making ISDN calls; one of "enbloc" or "overlap".
Interdigit	The maximum length of time, in seconds between digits before an unavailable indication is generated.
Data	Definition of the data terminating character.
Cadence	The cadence settings for each of the tones.
Bell	The cadencing for the bell tone.
Unavailable	The cadencing for the unavailable tone.

Examples To display the configuration of the PBX module, use the command:

```
SHOW PBX
```

See Also SET PBX

SHOW PBX CALL

Syntax SHOW PBX CALL

Description This command displays information about the state and duration of all active internal or external calls (Figure 14-5 on page 14-26, Table 14-5 on page 14-26). For outgoing external calls the called number is displayed. For incoming calls the calling number (if available from the Q.931 setup message) is displayed.

Figure 14-5: Example output from the SHOW PBX CALL command.

Orig. number	Dest. number	Held Number	State	Duration
-----	-----	-----	-----	-----
Exten 1	Exten 0	33778906	connected	0:00:09
-----	-----	-----	-----	-----

Table 14-5: Parameters displayed in the output of the SHOW PBX CALL command.

Parameter	Meaning
Orig. number	The originating external number or extension for the call.
Dest. number	The destination external number or extension for the call.
Held Number	The held external number or extension for the call (if any).
State	The state of the call; one of "connected" or "alerting".

Table 14-5: Parameters displayed in the output of the SHOW PBX CALL command.

Parameter	Meaning
Duration	The duration of the call in hours, minutes and seconds.

Examples To display information about PBX calls, use the command:

```
SHOW PBX CALLS
```

SHOW PBX EXTENSION

Syntax `SHOW PBX EXTENSION [=extension-number]`

where:

- *extension-number* is a string of decimal digits (0–9), 1 to 3 characters in length.

Description This command displays information about the extensions configured in the PBX module. If an extension is specified, information about that extension is displayed. Otherwise, information about all extensions is displayed (Figure 14-6 on page 14-27, Table 14-6 on page 14-28).

Figure 14-6: Example output from the SHOW PBX EXTENSION command.

```
Extension: 1
Type ..... user configured
Name ..... Extension 1
Port ..... 1
Suppress ..... 0
Auto terminate ..... 0
Group ..... default
Accept number ..... 2
Accept subaddress ..... off
Calling number ..... 3890123
Bearer Cap ..... speech
HLC ..... fax
No HLC ..... reject
Call waiting ..... on
```

Table 14-6: Parameters displayed in the output of the SHOW PBX EXTENSION command.

Parameter	Meaning
Extension	The number of the extension.
Type	The type of the extension; one of "system configured" for default extensions created by the system, or "user configured" if the extension has been created or modified by user commands.
Name	The text name assigned to the extension.
Port	The PBX port number of the extension, or "none" if the extension is not currently associated with a physical PBX port.
Suppress	The timer value, in seconds, for delaying the application of unavailable tone after the remote end of a connected call hangs up.
Auto terminate	The timer interval, in seconds, for the auto-termination feature for enbloc dialling.
Group	The PBX extension group to which this extension belongs, or "none" if the extension does not belong to a group.
Accept number	The number to match against the called number IE in the call setup message of incoming calls to determine whether or not to accept the call, or one of "all" (accept all calls), "not present" (accept calls with no called number IE) or "none" (do not accept any calls).
Accept Subaddress	The number to match against the subaddress IE in the call setup message of incoming calls to determine whether or not to accept the call, or one of "all" (accept all calls), "not present" (accept calls with no subaddress IE) or "none" (do not accept any calls).
Calling Number	The calling number for the extension, or "off" if no calling number has been configured.
Bearer Cap	The quality of circuit this extension will request from ISDN; one of "speech" or "3.1kHz audio".
HLC	The contents of the Bearer Capability IE, Low Layer Compatibility (LLC) IE and High Layer Compatibility (HLC) IE in call SETUP messages for calls made from this extension; one of "default", "telephone" or "fax".
No HLC	Whether or not to accept or reject incoming calls which do not have the High Layer Compatibility IE present in the SETUP message; one of "accept" or "reject".
Call waiting	The state of the call waiting feature; one of "on" (notify waiting calls) or "off" (do not notify waiting calls).

Examples To display information about extension 0, use the command:

```
SHOW PBX EXTENSION=0
```

See Also CREATE PBX EXTENSION
DESTROY PBX EXTENSION
SET PBX EXTENSION

SHOW PBX GROUP

Syntax `SHOW PBX GROUP [=group-name]`

where:

- *group-name* is a string, 1 to 15 characters in length. Valid characters are uppercase letters (A–Z), lowercase letters (a–z), digits (0–9) and spaces. If *group-name* contains spaces it must be enclosed in double quotes.

Description This command displays information about all extension groups configured in the PBX module. If a group is specified, only information about that group is displayed. Otherwise, information about all groups is displayed (Figure 14-7 on page 14-29, Table 14-7 on page 14-29).

Figure 14-7: Example output from the SHOW PBX GROUP command.

```
Group: Office
Extension ..... 2
Accept number ..... not present
Accept subaddress ..... off
Hunt ..... none

Extension List ..... 1      0
```

Table 14-7: Parameters displayed in the output of the SHOW PBX GROUP command.

Parameter	Meaning
Group	The text name assigned to the group.
Extension	The extension number assigned to the group.
Accept number	The number to match against the called number IE in the call setup message of incoming calls to determine whether or not to accept the call for the group, or one of "all" (accept all calls), "not present" (accept calls with no called number IE) or "none" (do not accept any calls).
Subaddress number	The number to match against the subaddress IE in the call setup message of incoming calls to determine whether or not to accept the call for the group, or one of "all" (accept all calls), "not present" (accept calls with no subaddress IE) or "none" (do not accept any calls).
Hunt	The type of hunting in use; one of "search" or "none".
Extension List	The list of extensions assigned to the group.

Examples To display information about the PBX group "Sales", use the command:

```
SHOW PBX GROUP="Sales"
```

See Also SET PBX GROUP

Chapter 15

Dynamic Host Configuration Protocol (DHCP)

Introduction	15-2
The Dynamic Host Configuration Protocol (DHCP)	15-2
Configuration Example	15-3
Command Reference	15-4
ADD DHCP POLICY	15-4
ADD DHCP RANGE	15-9
CREATE DHCP POLICY	15-9
CREATE DHCP RANGE	15-10
DELETE DHCP POLICY	15-11
DELETE DHCP RANGE	15-15
DESTROY DHCP POLICY	15-15
DESTROY DHCP RANGE	15-16
DISABLE DHCP	15-16
ENABLE DHCP	15-17
SET DHCP POLICY	15-17
SHOW DHCP	15-22
SHOW DHCP CLIENT	15-23
SHOW DHCP POLICY	15-24
SHOW DHCP RANGE	15-25

Introduction

This chapter describes the Dynamic Host Configuration Protocol (DHCP) support provided by the router, and how to configure the router to act as a DHCP or BOOTP server.

The Dynamic Host Configuration Protocol (DHCP) provides a method for passing configuration information to hosts on a TCP/IP network. DHCP is based on its predecessor Bootstrap Protocol (BOOTP), but adds automatic allocation of reusable network addresses and additional configuration options.

The Dynamic Host Configuration Protocol (DHCP)

The Dynamic Host Configuration Protocol (DHCP) is defined in RFC 1541 and provides a mechanism for passing configuration information to hosts on a TCP/IP network. DHCP is based on the Bootstrap Protocol (BOOTP) defined in RFC 1542, but adds automatic allocation of reusable network addresses and additional configuration options. DHCP is based on a client-server model, where the server is the host that allocates network addresses and initialisation parameters, and the client is the host that requests these parameters from the server.

DHCP supports three mechanisms for IP address allocation. In the *automatic allocation* mechanism, DHCP assigns a permanent IP address to a host. In the *dynamic allocation* mechanism, DHCP assigns an IP address to a host for a limited period of time, or until the host explicitly relinquishes the address. In the *manual allocation* mechanism, a host's IP address is assigned by the network administrator, and DHCP is used simply to convey the assigned address to the host. A particular network will use one or more of these mechanisms, depending on the policies of the network administrator.

Dynamic allocation is the only one of the three mechanisms that allows automatic reuse of an address that is no longer needed by the host to which it was assigned. Dynamic allocation is particularly useful for assigning an address to a host that will be connected to the network only temporarily, or for sharing a limited pool of IP addresses among a group of hosts that do not need permanent IP addresses. Dynamic allocation may also be a good choice for assigning an IP address to a new host being permanently connected to a network where IP addresses are sufficiently scarce that it is important to reclaim them when old hosts are retired. Manual allocation allows DHCP to be used to eliminate the error-prone process of manually configuring hosts with IP addresses in environments where (for whatever reasons) it is desirable to manage IP address assignment outside of the DHCP mechanisms.

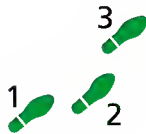
For dynamic allocation, DHCP assigns an IP address to a host for a limited period of time called the *lease time*. The minimum lease time is 3600 seconds. The maximum lease time is the largest unsigned 32-bit integer, called INFINITY in this implementation. If a lease time is set to INFINITY the mechanism changes to automatic allocation as the lease never expires.

This implementation uses the terms *policy* to refer to a predefined set of configuration information items, and the term *range* to refer to a list of consecutively numbered IP addresses.

This implementation supports both DHCP and its predecessor BOOTP, but this support must be explicitly enabled by a manager command. BOOTP requests can only be satisfied by policies with leases set to INFINITY, i.e. using the automatic allocation mechanism.

Configuration Example

The following example illustrates how to configure the router to act as a DHCP server in a small site. The site has a limited range of IP addresses and the users only use IP for short periods of time. The dynamic DHCP mechanism is the most appropriate for this situation. The router on the LAN will be configured to provide DHCP services to the PCs on the local LAN.



To configure DHCP:

1. Enable the DHCP Server.

To enable DHCP use the command:

```
ENABLE DHCP
```

2. Create a policy.

A policy is created setting the base configuration information required by the client hosts, using the commands:

```
CREATE DHCP POLICY=base LEASE=7200
ADD DHCP POLICY=base SUBNET=255.255.255.0
ADD DHCP POLICY=base ROUTER=192.168.1.1
ADD DHCP POLICY=base DNSSERVER=192.168.1.254,
192.168.1.253
```

3. Create a range.

Create a range that defines the list of IP addresses to which the policy applies, using the command:

```
CREATE DHCP RANGE=office POLICY=base IP=192.168.1.16
NUMBER=32
```

4. Test the configuration.

Check that DHCP is functioning correctly, using the commands:

```
SHOW DHCP
SHOW DHCP POLICY
SHOW DHCP RANGE
SHOW DHCP CLIENT
```

5. Configure a printer.

To configure a printer with the MAC address of 00-00-0c-00-28-73 that only talks BOOTP, use the commands:

```
ENABLE DHCP BOOTP
CREATE DHCP POLICY=prnt LEASE=INFINITY INHERIT=base
ADD DHCP RANGE=office POLICY=prnt IP=192.168.1.31
ADDRESS=00-00-0c-00-28-73
```

Command Reference

This section describes the commands available on the router to configure and manage the Dynamic Host Configuration Protocol (DHCP) on the router.

DHCP requires the IP module to be enabled and configured correctly. See *Chapter 6, Internet Protocol (IP)* for detailed descriptions of the commands required to enable and configure IP.

See “Conventions” on page xxxv of *Preface* in the front of this manual for details of the conventions used to describe command syntax. See *Appendix A, Messages* for a complete list of messages and their meanings.

ADD DHCP POLICY

Syntax `ADD DHCP POLICY=name [ARPTIMEOUT=seconds]
 [BOOTFILESIZE=bootfilesize] [BROADCASTADDRESS=ipadd]
 [COOKIESERVER=ipadd, ipadd...]
 [DNSSERVER=ipadd, ipadd...] [DOMAINNAME=string]
 [ETHERENCAP={ON|OFF}] [EXTENSIONPATH=string]
 [FILE=string] [HOSTNAME=string]
 [IMPRESSSERVER=ipadd, ipadd...] [INTMTU=mtu]
 [IPFORWARDING={ENABLED|DISABLED}] [IPMTU=mtu]
 [IPPLATEAU=mtu, mtu...] [IPTIMEOUT=seconds] [IPTTL=t1]
 [LOGSERVER=ipadd, ipadd...] [LPRSERVER=ipadd, ipadd...]
 [MASKDISCOVERY={ON|OFF}] [MASKSUPPLIER={ON|OFF}]
 [MERITDUMPFIL=string] [NAMESERVER=ipadd, ipadd...]
 [NBDDSERVERS=ipadd, ipadd...]
 [NBNAMESEVER=ipadd, ipadd...] [NBNODETYPE={BNODE|
 PNODE|MNODE|HNODE}] [NBSCOPE=string] [NISDOMAIN=string]
 [NISERVERS=ipadd, ipadd...] [NTPSERVERS=ipadd, ipadd...]
 [POLICYFILTERING=ipadd, ipadd...]
 [RESOURCESERVER=ipadd, ipadd...] [ROOTPATH=string]
 [ROUTER=ipadd, ipadd...] [ROUTERDISCOVERY={ON|OFF}]
 [ROUTERSOLICIT=ipadd] [SERVER=ipadd]
 [SERVERNAME=server-name] [SOURCEROUTING={ENABLED|
 DISABLED}] [STATICROUTE=ipadd, ipadd...] [SUBLOCAL={ON|
 OFF}] [SUBNETMASK=ipadd] [SWAPSERVER=ipadd]
 [T1TIME=seconds] [T2TIME=seconds] [TCPGARBAGE={ON|OFF}]
 [TCPKEEPALIVE=seconds] [TCPTTL=t1]
 [TIMEOFFSET=utc-offset] [TIMESERVER=ipadd, ipadd...]
 [TRAILERENCAP={ON|OFF}]
 [XDISPLAYSERVERS=ipadd, ipadd...]
 [XFONTSERVERS=ipadd, ipadd...]`

where:

- *name* is a character string, 1 to 15 characters in length. It may contain any printable character.
- *seconds* is a time, time offset or timeout value in seconds.
- *bootfilesize* is the length in 512-octet blocks of the default boot image for the client.
- *ipadd* is an IP address in dotted decimal notation.

- *string* is a character string, 1 to 99 characters in length. It may contain any printable character.
- *mtu* is the maximum size datagram that the client should be prepared to reassemble. The minimum value is 576.
- *tll* is a number in the range 1 and 255.
- *server-name* is a character string, 1 to 63 characters in length. It may contain any printable character.
- *utc-offset* is a time offset in seconds from Coordinated Universal Time (UTC).

Description This command adds an option to an existing DHCP policy. The POLICY parameter specifies the name of the policy to which the option is to be added.

The ARPTIMEOUT parameter specifies the timeout in seconds for ARP cache entries.

The BOOTFILESIZE parameter specifies the length in 512-octet blocks of the default boot image for the client.

The BROADCASTADDRESS parameter specifies the broadcast address in use on the client's subnet.

The COOKIESERVER parameter specifies a list of RFC 865 cookie servers available to the client. Servers should be listed in order of preference.

The DNSSERVER parameter specifies a list of Domain Name System (RFC 1035) name servers available to the client. Servers should be listed in order of preference.

The DOMAINNAME parameter specifies the domain name that client should use when resolving host names via the Domain Name System.

The ETHERENCAP parameter specifies whether or not the client should use Ethernet Version 2 (RFC 894) or IEEE 802.3 (RFC 1042) encapsulation if the interface is an Ethernet. A value of OFF indicates that the client should use RFC 894 encapsulation. A value of ON means that the client should use RFC 1042 encapsulation.

The EXTENSIONPATH parameter specifies a string to specify a file, retrievable via TFTP, which contains information which can be interpreted in the same way as the 64-octet vendor extension field within the BOOTP response.

The FILE parameter specifies the boot file name for the client.

The HOSTNAME parameter specifies the name of the client. The name may or may not be qualified with the local domain name. See RFC 1035 for character set restrictions.

The IMPRESSSERVER parameter specifies a list of Imagen Impress servers available to the client. Servers should be listed in order of preference.

The INTMTU parameter specifies the MTU to use on this interface. The MTU is specified as a 16-bit unsigned integer. The minimum legal value for the MTU is 68.

The IPFORWARDING parameter specifies whether or not the client should configure its IP layer for packet forwarding. A value of DISABLE will disable IP forwarding, and a value of ENABLE will enable IP forwarding.

The IPMTU parameter specifies the maximum size datagram that the client should be prepared to reassemble. The minimum value legal value is 576.

The IPPLATEAU parameter specifies a table of MTU sizes to use when performing Path MTU Discovery as defined in RFC 1191. The table is formatted as a list of 16-bit unsigned integers, ordered from smallest to largest. The minimum MTU value can not be smaller than 68.

The IPTIMEOUT parameter specifies the timeout (in seconds) to use when aging Path MTU values discovered by the mechanism defined in RFC 1191

The IPTTL parameter specifies the default time-to-live that the client should use on outgoing datagrams. The TTL is specified as an octet with a value between 1 and 255.

The LOGSERVER parameter specifies a list of MIT-LCS UDP log servers available to the client. Servers should be listed in order of preference.

The LPRSERVER parameter specifies a list of RFC 1179 line printer servers available to the client. Servers should be listed in order of preference.

The MASKDISCOVERY parameter specifies whether or not the client should perform subnet mask discovery using ICMP. A value of OFF indicates that the client should not perform mask discovery. A value of ON means that the client should perform mask discovery.

The MASKSUPPLIER parameter specifies whether or not the client should respond to subnet mask requests using ICMP. A value of OFF indicates that the client should not respond. A value of ON means that the client should respond.

The MERITDUMPFILe parameter specifies the path name of a file to which the client's core image should be dumped in the event the client crashes. The path name is formatted as a character string consisting of characters from the NVT ASCII character set.

The NAMESERVER parameter specifies a list of IEN116 name servers available to the client. Servers should be listed in order of preference.

The NBDDSERVERS parameter specifies a list of RFC 1001/1002 NetBIOS datagram distribution servers (NBDD) listed in order of preference.

The NBNAMESEVERs parameter specifies a list of RFC 1001/1002 NetBIOS name servers (NBNS) listed in order of preference.

The NBNODETYPE parameter specifies the NetBIOS node type which allows NetBIOS over TCP/IP clients to be configured as described in RFC 1001/1002.

The NBSCOPE parameter specifies the NetBIOS over TCP/IP scope parameter for the client as specified in RFC 1001/1002.

The NISDOMAIN parameter specifies the name of the client's NIS domain. The domain is formatted as a character string consisting of characters from the NVT ASCII character set.

The NISERVERS parameter specifies a list of IP addresses indicating NIS servers available to the client. Servers should be listed in order of preference.

The NTPSERVERS parameter specifies a list of IP addresses indicating NTP servers available to the client. Servers should be listed in order of preference.

The POLICYFILTERING parameter specifies policy filters for non-local source routing. The filters consist of a list of IP addresses and masks which specify destination/mask pairs with which to filter incoming source routes. Any source-routed datagram whose next hop address does not match one of the filters should be discarded by the client.

The RESOURCESERVER parameter specifies a list of RFC 887 Resource Location servers available to the client. Servers should be listed in order of preference.

The ROOTPATH parameter specifies the path name that contains the client's root disk. The path name is formatted as a character string consisting of characters from the NVT ASCII character set.

The ROUTER parameter specifies a list of IP addresses for routers on the client's subnet. Routers should be listed in order of preference.

The ROUTERDISCOVERY parameter specifies whether or not the client should solicit routers using the Router Discovery mechanism defined in RFC 1256. A value of OFF indicates that the client should not perform router discovery. A value of ON means that the client should perform router discovery.

The ROUTERSOLICIT parameter specifies the address to which the client should transmit router solicitation requests.

The SERVER parameter specifies the address of the server to use in the next step of the client's bootstrap process. As the router is not capable of providing an operating system executable this option allows the IP address of an appropriate TFTP server to be set.

The SERVERNAME parameter specifies the name of the server host. This is passed to the client.

The SOURCEROUTING parameter specifies whether or not the client should configure its IP layer to allow forwarding of datagrams with non-local source routes. A value of DISABLE will disallow forwarding of such datagrams, and a value of ENABLE will allow forwarding.

The STATICROUTE parameter specifies a list of static routes that the client should install in its routing cache. If multiple routes to the same destination are specified, they are listed in descending order of priority. The routes consist of a list of IP address pairs. The first address is the destination address, and the second address is the router for the destination. The default route (0.0.0.0) is an illegal destination for a static route.

The SUBLOCAL parameter specifies whether or not the client may assume that all subnets of the IP network to which the client is connected use the same MTU as the subnet of that network to which the client is directly connected. A value of ON indicates that all subnets share the same MTU. A value of OFF means that the client should assume that some subnets of the directly connected network may have smaller MTUs.

The SUBNETMASK parameter specifies the client's subnet mask as defined in RFC 950.

The SWAPSERVER parameter specifies the IP address of the client's swap server.

The T1TIME parameter specifies the time interval, in seconds, from address assignment until the client transitions to the RENEWING state.

The T2TIME parameter specifies the time interval, in seconds, from address assignment until the client transitions to the REBINDING state.

The TCPGARBAGE parameter specifies whether or not the client should send TCP keepalive messages with a octet of garbage for compatibility with older implementations. A value of OFF indicates that a garbage octet should not be sent. A value of ON indicates that a garbage octet should be sent.

The TCPKEEPALIVE parameter specifies the interval (in seconds) that the client TCP should wait before sending a keepalive message on a TCP connection. A value of zero indicates that the client should not generate keepalive messages on connections unless specifically requested by an application.

The TCPTTL parameter specifies the default time-to-live value that the client should use when sending TCP segments.

The TIMEOFFSET parameter specifies the offset of the client's subnet in seconds from Coordinated Universal Time (UTC).

The TIMESERVER parameter specifies a list of RFC 868 time servers available to the client. Servers should be listed in order of preference.

The TRAILERENCAP parameter specifies whether or not the client should negotiate the use of trailers (RFC 893) when using the ARP protocol. A value of OFF indicates that the client should not attempt to use trailers. A value of ON means that the client should attempt to use trailers.

The XDISPLAYSERVERS parameter specifies a list of IP addresses of systems that are running the X Window System Display Manager and are available to the client. Addresses should be listed in order of preference.

The XFONTSERVERS parameter specifies a list of X Window System Font servers available to the client. Servers should be listed in order of preference.

Examples To create a policy called "base" with subnet mask, router and DNS server options, use the command:

```
ADD DHCP POLICY=BASE SUBNETMASK=255.255.255.0
ROUTER=202.36.163.21
DNSSERVER=192.168.100.50,192.168.100.33
```

See Also CREATE DHCP POLICY
DELETE DHCP POLICY
DESTROY DHCP POLICY
SET DHCP POLICY
SHOW DHCP POLICY

ADD DHCP RANGE

Syntax `ADD DHCP RANGE=name IP=ipadd ADDRESS=macadd [POLICY=name]`

where:

- *name* is a character string, 1 to 15 characters in length. It may contain any printable character.
- *ipadd* is an IP address in dotted decimal notation.
- *macadd* is a hardware address of the form `xx-xx-xx-xx-xx-xx`, where `xx` is a two-digit hexadecimal number with leading zeros if necessary.

Description This command adds a static entry to an existing DHCP range. The RANGE parameter specifies the name of an existing DHCP range.

The IP parameter specifies the IP address of the host to add to the range. The ADDRESS parameter defines the MAC address for the static host entry. The POLICY parameter specifies the name of a policy to give the host entry.

Examples To add a static entry to the range “remote” for the device with MAC address 00-00-0c-00-28-73, use the command:

```
ADD DHCP RANGE=REMOTE IP=192.168.1.31
ADDRESS=00-00-0c-00-28-73
```

See Also CREATE DHCP RANGE
DELETE DHCP RANGE
DESTROY DHCP RANGE
SHOW DHCP RANGE

CREATE DHCP POLICY

Syntax `CREATE DHCP POLICY=name LEASETIME={lease-time|INFINITY}
[INHERIT=name]`

where:

- *name* is a character string, 1 to 15 characters in length. It may contain any printable character.
- *lease-time* is a time in seconds.

Description This command creates a DHCP policy. Policies define the configuration information that will be given to the requesting IP host. The POLICY parameter specifies the name of the policy to create. This name is used in other commands to identify the policy.

The LEASETIME parameter specifies the time period for which the IP address will be leased to the requesting IP client. For BOOTP requests this must be set to INFINITY. If dynamic IP address allocation is not required then set LEASETIME to INFINITY. The minimum value for LEASETIME is 3600 seconds.

The INHERIT parameter specifies the name of an existing policy whose settings will be inherited by the new policy. This parameter allows the building of

hierarchical policies and reduces the number of commands to create similar policies.

Examples To create a DHCP policy called “base” with a default lease time of two hours, use the command:

```
CREATE DHCP POLICY=base LEASE=7200
```

See Also ADD DHCP POLICY
DELETE DHCP POLICY
DESTROY DHCP POLICY
SET DHCP POLICY
SHOW DHCP POLICY

CREATE DHCP RANGE

Syntax CREATE DHCP RANGE=*name* POLICY=*name* IP=*ipadd* NUMBER=*number*
[GATEWAY=*ipadd*]

where:

- *name* is a character string, 1 to 15 characters in length. It may contain any printable character.
- *ipadd* is an IP address in dotted decimal notation.
- *number* is a number in the range 1 to 255.

Description This command creates a DHCP range. The server will reply try to fulfil BOOTP or DHCP requests from hosts with IP addresses in the defined ranges. The RANGE parameter specifies the name of the range to create.

The POLICY parameter specifies the name of a default policy to give the range. Individual host entries in the range can later be set to other defined policies.

The IP address parameter defines the IP address of the start of the range.

The NUMBER parameter defines how many host entries from the start IP address are to be included in the range.

The GATEWAY parameter specifies the IP address of a remote BOOTP relay agent. This parameter is needed if the range of IP addresses specified are not on a local interface.

Examples To create a range called “office”, which uses the policy “base”, with 32 IP addresses starting at 192.168.1.16, use the command:

```
CREATE DHCP RANGE=office POLICY=base IP=192.168.1.16  
NUMBER=32
```

See Also ADD DHCP RANGE
DELETE DHCP RANGE
DESTROY DHCP RANGE
SHOW DHCP RANGE

DELETE DHCP POLICY

Syntax DELETE DHCP POLICY=*name* [ARPTIMEOUT] [BOOTFILESIZE]
 [BROADCASTADDRESS] [COOKIESERVER] [DNSSERVER]
 [DOMAINNAME] [ETHERENCAP] [EXTENSIONPATH] [FILE]
 [HOSTNAME] [IMPRESSSERVER] [INTMTU] [IPFORWARDING]
 [IPMTU] [IPPLATEAU] [IPTIMEOUT] [IPTTL] [LOGSERVER]
 [LPRSERVER] [MASKDISCOVERY] [MASKSUPPLIER]
 [MERITDUMPFIL] [NAMESERVER] [NBDDSERVERS]
 [NBNAMESEVERES] [NBNODETYPE] [NBSCOPE] [NISDOMAIN]
 [NISERVERS] [NTPSERVERS] [POLICYFILTERING]
 [RESOURCESEVER] [ROOTPATH] [ROUTER] [ROUTERDISCOVERY]
 [ROUTERSOLICIT] [SERVER] [SERVERNAME] [SOURCEROUTING]
 [STATICROUTE] [SUBLOCAL] [SUBNETMASK] [SWAPSERVER]
 [T1TIME] [T2TIME] [TCPGARBAGE] [TCPKEEPALIVE] [TCPPTTL]
 [TIMEOFFSET] [TIMESERVER] [TRAILERENCAP]
 [XDISPLAYSERVERS] [XFONTSERVERS]

where:

- *name* is a character string, 1 to 15 characters in length. It may contain any printable character.

Description This command deletes an existing option from a DHCP policy. The POLICY parameter specifies the name of the policy from which the option is to be deleted.

The ARPTIMEOUT parameter specifies the timeout in seconds for ARP cache entries.

The BOOTFILESIZE parameter specifies the length in 512-octet blocks of the default boot image for the client.

The BROADCASTADDRESS parameter specifies the broadcast address in use on the client's subnet.

The COOKIESERVER parameter specifies a list of RFC 865 cookie servers available to the client. Servers should be listed in order of preference.

The DNSSERVER parameter specifies a list of Domain Name System (RFC 1035) name servers available to the client. Servers should be listed in order of preference.

The DOMAINNAME parameter specifies the domain name that client should use when resolving hostnames via the Domain Name System.

The ETHERENCAP parameter specifies whether or not the client should use Ethernet Version 2 (RFC 894) or IEEE 802.3 (RFC 1042) encapsulation if the interface is an Ethernet. A value of OFF indicates that the client should use RFC 894 encapsulation. A value of ON means that the client should use RFC 1042 encapsulation.

The EXTENSIONPATH parameter specifies a string to specify a file, retrievable via TFTP, which contains information which can be interpreted in the same way as the 64-octet vendor -extension field within the BOOTP response.

The FILE parameter specifies the boot file name for the client.

The HOSTNAME parameter specifies the name of the client. The name may or may not be qualified with the local domain name. See RFC 1035 for character set restrictions.

The IMPRESSSERVER parameter specifies a list of Imagen Impress servers available to the client. Servers should be listed in order of preference.

The INTMTU parameter specifies the MTU to use on this interface. The MTU is specified as a 16-bit unsigned integer. The minimum legal value for the MTU is 68.

The IPFORWARDING parameter specifies whether or not the client should configure its IP layer for packet forwarding. A value of DISABLE will disable IP forwarding, and a value of ENABLE will enable IP forwarding.

The IPMTU parameter specifies the maximum size datagram that the client should be prepared to reassemble. The minimum value legal value is 576.

The IPPLATEAU parameter specifies a table of MTU sizes to use when performing Path MTU Discovery as defined in RFC 1191. The table is formatted as a list of 16-bit unsigned integers, ordered from smallest to largest. The minimum MTU value can not be smaller than 68.

The IPTIMEOUT parameter specifies the timeout (in seconds) to use when aging Path MTU values discovered by the mechanism defined in RFC1191

The IPTTL parameter specifies the default time-to-live that the client should use on outgoing datagrams. The TTL is specified as an octet with a value between 1 and 255.

The LOGSERVER parameter specifies a list of MIT-LCS UDP log servers available to the client. Servers should be listed in order of preference.

The LPRSERVER parameter specifies a list of RFC 1179 line printer servers available to the client. Servers should be listed in order of preference.

The MASKDISCOVERY parameter specifies whether or not the client should perform subnet mask discovery using ICMP. A value of OFF indicates that the client should not perform mask discovery. A value of ON means that the client should perform mask discovery.

The MASKSUPPLIER parameter specifies whether or not the client should respond to subnet mask requests using ICMP. A value of OFF indicates that the client should not respond. A value of ON means that the client should respond.

The MERITDUMPFILe parameter specifies the path name of a file to which the client's core image should be dumped in the event the client crashes. The path name is formatted as a character string consisting of characters from the NVT ASCII character set.

The NAMESERVER parameter specifies a list of IEN116 name servers available to the client. Servers should be listed in order of preference.

The NBDDSERVERS parameter specifies a list of RFC 1001/1002 NetBIOS datagram distribution servers (NBDD) listed in order of preference.

The NBNAMESEVERs parameter specifies a list of RFC 1001/1002 NetBIOS name servers (NBNS) listed in order of preference.

The NBNODETYPE parameter specifies the NetBIOS node type which allows NetBIOS over TCP/IP clients to be configured as described in RFC 1001/1002.

The NBSCOPE parameter specifies the NetBIOS over TCP/IP scope parameter for the client as specified in RFC 1001/1002.

The NISDOMAIN parameter specifies the name of the client's NIS domain. The domain is formatted as a character string consisting of characters from the NVT ASCII character set.

The NISERVERS parameter specifies a list of IP addresses indicating NIS servers available to the client. Servers should be listed in order of preference.

The NTPSERVERS parameter specifies a list of IP addresses indicating NTP servers available to the client. Servers should be listed in order of preference.

The POLICYFILTERING parameter specifies policy filters for non-local source routing. The filters consist of a list of IP addresses and masks which specify destination/mask pairs with which to filter incoming source routes. Any source-routed datagram whose next hop address does not match one of the filters should be discarded by the client.

The RESOURCESERVER parameter specifies a list of RFC 887 Resource Location servers available to the client. Servers should be listed in order of preference.

The ROOTPATH parameter specifies the path name that contains the client's root disk. The path name is formatted as a character string consisting of characters from the NVT ASCII character set.

The ROUTER parameter specifies a list of IP addresses for routers on the client's subnet. Routers should be listed in order of preference.

The ROUTERDISCOVERY parameter specifies whether or not the client should solicit routers using the Router Discovery mechanism defined in RFC 1256. A value of OFF indicates that the client should not perform router discovery. A value of ON means that the client should perform router discovery.

The ROUTERSOLICIT parameter specifies the address to which the client should transmit router solicitation requests.

The SERVER parameter specifies the address of the server to use in the next step of the client's bootstrap process. As the router is not capable of providing an operating system executable this option allows the IP address of an appropriate TFTP server to be set.

The SERVERNAME parameter specifies the name of the server host. This is passed to the client.

The SOURCEROUTING parameter specifies whether or not the client should configure its IP layer to allow forwarding of datagrams with non-local source routes. A value of DISABLE will disallow forwarding of such datagrams, and a value of ENABLE will allow forwarding.

The STATICROUTE parameter specifies a list of static routes that the client should install in its routing cache. If multiple routes to the same destination are specified, they are listed in descending order of priority. The routes consist of a list of IP address pairs. The first address is the destination address, and the

second address is the router for the destination. The default route (0.0.0.0) is an illegal destination for a static route.

The SUBLOCAL parameter specifies whether or not the client may assume that all subnets of the IP network to which the client is connected use the same MTU as the subnet of that network to which the client is directly connected. A value of ON indicates that all subnets share the same MTU. A value of OFF means that the client should assume that some subnets of the directly connected network may have smaller MTUs.

The SUBNETMASK parameter specifies the client's subnet mask as defined in RFC 950.

The SWAPSERVER parameter specifies the IP address of the client's swap server.

The T1TIME parameter specifies the time interval, in seconds, from address assignment until the client transitions to the RENEWING state.

The T2TIME parameter specifies the time interval, in seconds, from address assignment until the client transitions to the REBINDING state.

The TCPGARBAGE parameter specifies whether or not the client should send TCP keepalive messages with a octet of garbage for compatibility with older implementations. A value of OFF indicates that a garbage octet should not be sent. A value of ON indicates that a garbage octet should be sent.

The TCPKEEPALIVE parameter specifies the interval (in seconds) that the client TCP should wait before sending a keepalive message on a TCP connection. A value of zero indicates that the client should not generate keepalive messages on connections unless specifically requested by an application.

The TCPTTL parameter specifies the default time-to-live value that the client should use when sending TCP segments.

The TIMEOFFSET parameter specifies the offset of the client's subnet in seconds from Coordinated Universal Time (UTC).

The TIMESERVER parameter specifies a list of RFC 868 time servers available to the client. Servers should be listed in order of preference.

The TRAILERENCAP parameter specifies whether or not the client should negotiate the use of trailers (RFC 893) when using the ARP protocol. A value of OFF indicates that the client should not attempt to use trailers. A value of ON means that the client should attempt to use trailers.

The XDISPLAYSERVERS parameter specifies a list of IP addresses of systems that are running the X Window System Display Manager and are available to the client. Addresses should be listed in order of preference.

The XFONTSERVERS parameter specifies a list of X Window System Font servers available to the client. Servers should be listed in order of preference.

Examples To remove the LPRSERVER option from the policy "base", use the command:

```
DELETE DHCP POLICY=BASE LPRSERVER
```

See Also ADD DHCP POLICY
CREATE DHCP POLICY
DESTROY DHCP POLICY
SET DHCP POLICY
SHOW DHCP POLICY

DELETE DHCP RANGE

Syntax DELETE DHCP RANGE=*name* IP=*ipadd*

where:

- *name* is a character string, 1 to 15 characters in length. It may contain any printable character.
- *ipadd* is an IP address in dotted decimal notation.

Description This command deletes an existing static host entry from a DHCP range. The IP host entry reverts to the default settings for the range.

The RANGE parameter specifies the name of the range. The IP address parameter specifies the host entry to return to the default range settings.

Examples To delete the static entry 192.168.1.31 from the range "remote", use the command:

```
DELETE DHCP RANGE=REMOTE IP=192.168.1.31
```

See Also ADD DHCP RANGE
CREATE DHCP RANGE
DESTROY DHCP RANGE
SHOW DHCP RANGE

DESTROY DHCP POLICY

Syntax DESTROY DHCP POLICY=*name*

where:

- *name* is a character string, 1 to 15 characters in length. It may contain any printable character.

Description This command destroys an existing policy. The POLICY parameter specifies the name of the policy to destroy. If the policy is currently being used by any host entry, then an error message is displayed and the command fails.

Examples To destroy policy "admin", use the command:

```
DESTROY DHCP POLICY=ADMIN
```

See Also ADD DHCP POLICY
CREATE DHCP POLICY
DELETE DHCP POLICY
SET DHCP POLICY
SHOW DHCP POLICY

DESTROY DHCP RANGE

Syntax DESTROY DHCP RANGE=*name*

where:

- *name* is a character string, 1 to 15 characters in length. It may contain any printable character.

Description This command destroys an existing DHCP range. The RANGE parameter specifies the name of the range to destroy.

Examples To destroy the range "remote", use the command:

```
DESTROY DHCP RANGE=REMOTE
```

See Also ADD DHCP RANGE
CREATE DHCP RANGE
DELETE DHCP RANGE
SHOW DHCP RANGE

DISABLE DHCP

Syntax DISABLE DHCP [BOOTP]

Description This command disables the DHCP module. All BOOTP or DHCP requests received while the module is disabled are ignored.

If the optional parameter BOOTP is specified then only the reception of BOOTP requests is disabled.

Examples To disable BOOTP serving, use the command:

```
DISABLE DHCP BOOTP
```

See Also ENABLE DHCP
SHOW DHCP

ENABLE DHCP

Syntax `ENABLE DHCP [BOOTP]`

Description This command enables the DHCP module. All BOOTP or DHCP requests received while the module is disabled are ignored.

If the optional parameter BOOTP is specified then only the reception of BOOTP requests is enabled.

Examples To enable the DHCP server, use the command:

```
ENABLE DHCP
```

See Also `DISABLE DHCP`
`SHOW DHCP`

SET DHCP POLICY

Syntax `SET DHCP POLICY=name [ARPTIMEOUT=seconds]`
`[BOOTFILESIZE=bootfilesize] [BROADCASTADDRESS=ipadd]`
`[COOKIESERVER=ipadd,ipadd...]`
`[DNSSERVER=ipadd,ipadd...] [DOMAINNAME=string]`
`[ETHERENCAP={ON|OFF}] [EXTENSIONPATH=string]`
`[FILE=string] [HOSTNAME=string]`
`[IMPRESSSERVER=ipadd,ipadd...] [INTMTU=mtu]`
`[IPFORWARDING={ENABLED|DISABLED}] [IPMTU=mtu]`
`[IPPLATEAU=mtu,mtu...] [IPTIMEOUT=seconds] [IPTTL=t1]`
`[LOGSERVER=ipadd,ipadd...] [LPRSERVER=ipadd,ipadd...]`
`[MASKDISCOVERY={ON|OFF}] [MASKSUPPLIER={ON|OFF}]`
`[MERITDUMPFIL=string] [NAMESERVER=ipadd,ipadd...]`
`[NBDDSERVERS=ipadd,ipadd...]`
`[NBNAMESEVER=ipadd,ipadd...] [NBNODETYPE={BNODE|`
`PNODE|MNODE|HNODE}] [NBSCOPE=string] [NISDOMAIN=string]`
`[NISERVERS=ipadd,ipadd...] [NTPSERVERS=ipadd,ipadd...]`
`[POLICYFILTERING=ipadd,ipadd...]`
`[RESOURCESEVER=ipadd,ipadd...] [ROOTPATH=string]`
`[ROUTER=ipadd,ipadd...] [ROUTERDISCOVERY={ON|OFF}]`
`[ROUTERSOLICIT=ipadd] [SERVER=ipadd]`
`[SERVERNAME=server-name] [SOURCEROUTING={ENABLED|`
`DISABLED}] [STATICROUTE=ipadd,ipadd...] [SUBLOCAL={ON|`
`OFF}] [SUBNETMASK=ipadd] [SWAPSERVER=ipadd]`
`[T1TIME=seconds] [T2TIME=seconds] [TCPGARBAGE={ON|OFF}]`
`[TCPKEEPALIVE=seconds] [TCPTTL=t1]`
`[TIMEOFFSET=utc-offset] [TIMESERVER=ipadd,ipadd...]`
`[TRAILERENCAP={ON|OFF}]`
`[XDISPLAYSERVERS=ipadd,ipadd...]`
`[XFONTSERVERS=ipadd,ipadd...]`

where:

- *name* is a character string, 1 to 15 characters in length. It may contain any printable character.
- *seconds* is a time, time offset or timeout value in seconds.

- *bootfilesize* is the length in 512-octet blocks of the default boot image for the client.
- *ipadd* is an IP address in dotted decimal notation.
- *string* is a character string, 1 to 99 characters in length. It may contain any printable character.
- *mtu* is the maximum size datagram that the client should be prepared to reassemble. The minimum value is 576.
- *tth* is a number in the range 1 and 255.
- *server-name* is a character string, 1 to 63 characters in length. It may contain any printable character.
- *utc-offset* is a time offset in seconds from Coordinated Universal Time (UTC).

Description This command modifies an existing option in a DHCP policy. The POLICY parameter specifies the name of the policy containing the option to be modified.

The ARPTIMEOUT parameter specifies the timeout in seconds for ARP cache entries.

The BOOTFILESIZE parameter specifies the length in 512-octet blocks of the default boot image for the client.

The BROADCASTADDRESS parameter specifies the broadcast address in use on the client's subnet.

The COOKIESERVER parameter specifies a list of RFC 865 cookie servers available to the client. Servers should be listed in order of preference.

The DNSSERVER parameter specifies a list of Domain Name System (RFC 1035) name servers available to the client. Servers should be listed in order of preference.

The DOMAINNAME parameter specifies the domain name that client should use when resolving hostnames via the Domain Name System.

The ETHERENCAP parameter specifies whether or not the client should use Ethernet Version 2 (RFC 894) or IEEE 802.3 (RFC 1042) encapsulation if the interface is an Ethernet. A value of OFF indicates that the client should use RFC 894 encapsulation. A value of ON means that the client should use RFC 1042 encapsulation.

The EXTENSIONPATH parameter specifies a string to specify a file, retrievable via TFTP, which contains information which can be interpreted in the same way as the 64-octet vendor extension field within the BOOTP response.

The FILE parameter specifies the boot file name for the client.

The HOSTNAME parameter specifies the name of the client. The name may or may not be qualified with the local domain name. See RFC 1035 for character set restrictions.

The IMPRESSSERVER parameter specifies a list of Imagen Impress servers available to the client. Servers should be listed in order of preference.

The INTMTU parameter specifies the MTU to use on this interface. The MTU is specified as a 16-bit unsigned integer. The minimum legal value for the MTU is 68.

The IPFORWARDING parameter specifies whether or not the client should configure its IP layer for packet forwarding. A value of DISABLE will disable IP forwarding, and a value of ENABLE will enable IP forwarding.

The IPMTU parameter specifies the maximum size datagram that the client should be prepared to reassemble. The minimum value legal value is 576.

The IPPLATEAU parameter specifies a table of MTU sizes to use when performing Path MTU Discovery as defined in RFC 1191. The table is formatted as a list of 16-bit unsigned integers, ordered from smallest to largest. The minimum MTU value can not be smaller than 68.

The IPTIMEOUT parameter specifies the timeout (in seconds) to use when aging Path MTU values discovered by the mechanism defined in RFC1191

The IPTTL parameter specifies the default time-to-live that the client should use on outgoing datagrams. The TTL is specified as an octet with a value between 1 and 255.

The LOGSERVER parameter specifies a list of MIT-LCS UDP log servers available to the client. Servers should be listed in order of preference.

The LPRSERVER parameter specifies a list of RFC 1179 line printer servers available to the client. Servers should be listed in order of preference.

The MASKDISCOVERY parameter specifies whether or not the client should perform subnet mask discovery using ICMP. A value of OFF indicates that the client should not perform mask discovery. A value of ON means that the client should perform mask discovery.

The MASKSUPPLIER parameter specifies whether or not the client should respond to subnet mask requests using ICMP. A value of OFF indicates that the client should not respond. A value of ON means that the client should respond.

The MERITDUMPFILe parameter specifies the path name of a file to which the client's core image should be dumped in the event the client crashes. The path name is formatted as a character string consisting of characters from the NVT ASCII character set.

The NAMESERVER parameter specifies a list of IEN116 name servers available to the client. Servers should be listed in order of preference.

The NBDDSERVERS parameter specifies a list of RFC 1001/1002 NetBIOS datagram distribution servers (NBDD) listed in order of preference.

The NBNAMESEVERs parameter specifies a list of RFC 1001/1002 NetBIOS name servers (NBNS) listed in order of preference.

The NBNODETYPE parameter specifies the NetBIOS node type which allows NetBIOS over TCP/IP clients to be configured as described in RFC 1001/1002.

The NBSCOPE parameter specifies the NetBIOS over TCP/IP scope parameter for the client as specified in RFC 1001/1002.

The NISDOMAIN parameter specifies the name of the client's NIS domain. The domain is formatted as a character string consisting of characters from the NVT ASCII character set.

The NISERVERS parameter specifies a list of IP addresses indicating NIS servers available to the client. Servers should be listed in order of preference.

The NTPSERVERS parameter specifies a list of IP addresses indicating NTP servers available to the client. Servers should be listed in order of preference.

The POLICYFILTERING parameter specifies policy filters for non-local source routing. The filters consist of a list of IP addresses and masks which specify destination/mask pairs with which to filter incoming source routes. Any source-routed datagram whose next hop address does not match one of the filters should be discarded by the client.

The RESOURCESERVER parameter specifies a list of RFC 887 Resource Location servers available to the client. Servers should be listed in order of preference.

The ROOTPATH parameter specifies the path name that contains the client's root disk. The path name is formatted as a character string consisting of characters from the NVT ASCII character set.

The ROUTER parameter specifies a list of IP addresses for routers on the client's subnet. Routers should be listed in order of preference.

The ROUTERDISCOVERY parameter specifies whether or not the client should solicit routers using the Router Discovery mechanism defined in RFC 1256. A value of OFF indicates that the client should not perform router discovery. A value of ON means that the client should perform router discovery.

The ROUTERSOLICIT parameter specifies the address to which the client should transmit router solicitation requests.

The SERVER parameter specifies the address of the server to use in the next step of the client's bootstrap process. As the router is not capable of providing an operating system executable this option allows the IP address of an appropriate TFTP server to be set.

The SERVERNAME parameter specifies the name of the server host. This is passed to the client.

The SOURCEROUTING parameter specifies whether or not the client should configure its IP layer to allow forwarding of datagrams with non-local source routes. A value of DISABLE will disallow forwarding of such datagrams, and a value of ENABLE will allow forwarding.

The STATICROUTE parameter specifies a list of static routes that the client should install in its routing cache. If multiple routes to the same destination are specified, they are listed in descending order of priority. The routes consist of a list of IP address pairs. The first address is the destination address, and the second address is the router for the destination. The default route (0.0.0.0) is an illegal destination for a static route.

The SUBLOCAL parameter specifies whether or not the client may assume that all subnets of the IP network to which the client is connected use the same MTU as the subnet of that network to which the client is directly connected. A

value of ON indicates that all subnets share the same MTU. A value of OFF means that the client should assume that some subnets of the directly connected network may have smaller MTUs.

The SUBNETMASK parameter specifies the client's subnet mask as defined in RFC 950.

The SWAPSERVER parameter specifies the IP address of the client's swap server.

The T1TIME parameter specifies the time interval, in seconds, from address assignment until the client transitions to the RENEWING state.

The T2TIME parameter specifies the time interval, in seconds, from address assignment until the client transitions to the REBINDING state.

The TCPGARBAGE parameter specifies whether or not the client should send TCP keepalive messages with a octet of garbage for compatibility with older implementations. A value of OFF indicates that a garbage octet should not be sent. A value of ON indicates that a garbage octet should be sent.

The TCPKEEPALIVE parameter specifies the interval (in seconds) that the client TCP should wait before sending a keepalive message on a TCP connection. A value of zero indicates that the client should not generate keepalive messages on connections unless specifically requested by an application.

The TCPTTL parameter specifies the default time-to-live value that the client should use when sending TCP segments.

The TIMEOFFSET parameter specifies the offset of the client's subnet in seconds from Coordinated Universal Time (UTC).

The TIMESERVER parameter specifies a list of RFC 868 time servers available to the client. Servers should be listed in order of preference.

The TRAILERENCAP parameter specifies whether or not the client should negotiate the use of trailers (RFC 893) when using the ARP protocol. A value of OFF indicates that the client should not attempt to use trailers. A value of ON means that the client should attempt to use trailers.

The XDISPLAYSERVERS parameter specifies a list of IP addresses of systems that are running the X Window System Display Manager and are available to the client. Addresses should be listed in order of preference.

The XFONTSERVERS parameter specifies a list of X Window System Font servers available to the client. Servers should be listed in order of preference.

Examples To change the DN server for policy "base", use the command:

```
SET DHCP POLICY=BASE DNSSERVER=192.168.100.51
```

See Also ADD DHCP POLICY
CREATE DHCP POLICY
DELETE DHCP POLICY
DESTROY DHCP POLICY
SHOW DHCP POLICY

SHOW DHCP

Syntax SHOW DHCP

Description This command displays the state of the DHCP module (Figure 15-1 on page 15-22, Table 15-1 on page 15-22).

Figure 15-1: Example output from the SHOW DHCP command.

```
DHCP Server

State ..... enabled
BOOTP Status ..... enabled
Policies ..... poll
                prnt
Ranges ..... develop ( 202.36.163.6 - 202.36.163.22 )
                remote ( 192.168.100.92 - 192.168.100.124 )
In Messages ..... 3
Out Messages ..... 3
In DHCP Messages ..... 3
Out DHCP Messages ..... 3
In BOOTP Messages ..... 0
Out BOOTP Messages ..... 0
```

Table 15-1: Parameters displayed in the output of the SHOW DHCP command.

Parameter	Meaning
State	The status of the DHCP server; one of “enabled” or “disabled”.
BOOTP Status	The status of BOOTP serving; one of “enabled” or “disabled”.
Policies	A list of the policies that have been defined.
Ranges	A list of the ranges that have been defined.
In Messages	The total number of DHCP or BOOTP messages received by the router.
Out Messages	The total number of DHCP or BOOTP messages transmitted by the router.
In DHCP Messages	The number of DHCP messages received by the router.
Out DHCP Messages	The number of DHCP messages transmitted by the router.
In BOOTP Messages	The number of BOOTP messages received by the router.
Out BOOTP Messages	The number of BOOTP messages transmitted by the router.

Examples To display the current configuration of the DHCP server, use the command:

```
SHOW DHCP
```

See Also SHOW DHCP CLIENT
SHOW DHCP POLICY
SHOW DHCP RANGE

SHOW DHCP CLIENT

Syntax SHOW DHCP CLIENT [RANGE=*name*]

Description This command displays information about the currently defined range client entries (Figure 15-2 on page 15-23, Table 15-2 on page 15-24). If the RANGE parameter is specified then only the clients in the specified range are displayed.

Figure 15-2: Example output from the SHOW DHCP CLIENT command.

DHCP Client Entries				
IP Address	ClientId	State	Type	Expiry
202.36.163.14	00-00-c0-00-00-01	unused	static	never
202.36.163.15	00-00-c0-00-00-02	unused	static	never
202.36.163.16	00-00-c0-00-00-03	unused	static	never
202.36.163.17	00-00-c0-00-00-04	unused	static	never
202.36.163.18	00-00-c0-00-00-05	unused	static	never
202.36.163.19	00-00-c0-00-00-06	unused	static	never
202.36.163.20	08-00-5a-a1-02-3f	inuse	auto	never
202.36.163.21	00-00-c0-c9-c6-7b	inuse	auto	never
202.36.163.22	08-00-09-0d-16-e7	inuse	auto	never
202.36.163.23		unused	auto	never
202.36.163.24		unused	auto	never
202.36.163.25		unused	auto	never
202.36.163.26		unused	auto	never
202.36.163.27		unused	auto	never
202.36.163.28	00-40-10-02-e8-a3	inuse	auto	never
192.168.100.92	00-00-c0-c9-c6-21	inuse	dyn	19-Jun-1997 12:30:51
192.168.100.93		unused	dyn	
192.168.100.94		unused	dyn	
192.168.100.95		unused	dyn	
192.168.100.96		unused	dyn	
192.168.100.97		unused	dyn	
192.168.100.98		unused	dyn	
192.168.100.99		unused	dyn	
192.168.100.110		unused	dyn	
192.168.100.111		unused	dyn	
192.168.100.112		unused	dyn	
192.168.100.113		unused	dyn	
192.168.100.114		unused	dyn	
192.168.100.115		reclaim	dyn	
192.168.100.116		reclaim	dyn	
192.168.100.117		reclaim	dyn	
192.168.100.118		reclaim	dyn	

Table 15-2: Parameters displayed in the output of the SHOW DHCP CLIENT command.

Parameter	Meaning
IP Address	An IP address from the range of available IP addresses.
ClientId	The hardware address of the client, if any, that has been assigned the IP address.
State	The state of the IP address; one of "unused" (the IP address is not currently in use and is available for assignment), "inuse" (the IP address is currently assigned to a client) or "reclaim" (the IP address is currently being reclaimed).
Type	The type of allocation mechanism applied to the IP address; one of "static" (manual allocation), "auto" (automatic allocation) or "dyn" (dynamic allocation).
Expiry	The expiry date for dynamically allocated IP addresses.

Examples To display information about the clients in the range "remote", use the command:

```
SHOW DHCP CLIENT RANGE=REMOTE
```

See Also SHOW DHCP
SHOW DHCP POLICY
SHOW DHCP RANGE

SHOW DHCP POLICY

Syntax SHOW DHCP POLICY [=name]

Description This command displays information about the currently defined policies (Figure 15-3 on page 15-24, Table 15-3 on page 15-25). If a policy name is specified then only information about the specified policy is displayed.

Figure 15-3: Example output from the SHOW DHCP POLICY command.

```
DHCP Policies

Name: pol1
Base Policy: none
01 subnetmask ..... 255.255.255.0
03 router ..... 202.36.163.21
06 dnsserver ..... 192.168.100.50 192.168.100.33
51 leasetime ..... 3600

Name: prnt
Base Policy: pol1
01 subnetmask ..... (pol1) 255.255.255.0
03 router ..... (pol1) 202.36.163.21
06 dnsserver ..... (pol1) 192.168.100.50 192.168.100.33
51 leasetime ..... (prnt) infinity
```


Table 15-3: Parameters displayed in the output of the SHOW DHCP POLICY command.

Parameter	Meaning
Name	The name of the policy.
Base Policy	The base policy inherited by this policy.
options...	A list of the options configured for the policy. Each entry includes the DHCP option identifier, the parameter keyword and the current value(s) of the option.

Examples To display information about the policy “base”, use the command:

```
SHOW DHCP POLICY=BASE
```

See Also SHOW DHCP
SHOW DHCP CLIENT
SHOW DHCP RANGE

SHOW DHCP RANGE

Syntax SHOW DHCP RANGE [=name]

Description This command displays information about the currently defined ranges (Figure 15-4 on page 15-25, Table 15-4 on page 15-26). If a range name is specified then only information about the specified range is displayed.

Figure 15-4: Example output from the SHOW DHCP RANGE command.

```
DHCP Ranges

Name: remote
  Start Address ..... 192.168.100.92
  End Address ..... 192.168.100.124
  Used Address(es) ..... 192.168.100.92      192.168.100.94      192.168.100.95
                           192.168.100.96
  Free Address(es) ..... 192.168.100.93      192.168.100.97      192.168.100.98
                           192.168.100.99      192.168.100.100     192.168.100.101
                           192.168.100.102
  Reclaiming Address(es) ..... 192.168.100.103  192.168.100.104  192.168.100.105
                           192.168.100.106  192.168.100.107  192.168.100.108
                           192.168.100.109  192.168.100.110  192.168.100.111
                           192.168.100.112  192.168.100.113  192.168.100.114
                           192.168.100.115  192.168.100.116  192.168.100.117
                           192.168.100.118  192.168.100.119  192.168.100.120
                           192.168.100.121  192.168.100.122  192.168.100.123

  In DHCP Messages ..... 0
  In Discover Messages ..... 0
  In Request Messages ..... 0
  In Decline Messages ..... 0
  In Release Messages ..... 0
  Out DHCP Messages ..... 0
  Out Offer Messages ..... 0
  Out Ack Messages ..... 0
  Out Nak Messages ..... 0
  In BOOTP Messages ..... 0
  Out BOOTP Messages ..... 0
```

Table 15-4: Parameters displayed in the output of the SHOW DHCP RANGE command.

Parameter	Meaning
Name	The name of the range.
Start Address	The first IP address in the range.
End Address	The last IP address in the range.
Used Address(es)	A list of the IP addresses currently assigned to clients.
Free Address(es)	A list of the IP addresses currently available for assignment.
Reclaiming Address(es)	A list of the IP addresses currently being reclaimed from clients.
In DHCP Messages	The total number of DHCP messages received for this range.
In Discover Messages	The number of DHCP discover messages received for this range.
In Request Messages	The number of DHCP request messages received for this range.
In Decline Messages	The number of DHCP decline messages received for this range.
In Release Messages	The number of DHCP release messages received for this range.
Out DHCP Messages	The total number of DHCP messages transmitted for this range.
Out Offer Messages	The number of DHCP offer messages transmitted for this range.
Out Ack Messages	The number of DHCP acknowledgment (ACK) messages transmitted for this range.
Out Nak Messages	The number of DHCP negative acknowledgement (NAK) messages transmitted for this range.
In BOOTP Messages	The number of BOOTP messages received for this range.
Out BOOTP Messages	The number of BOOTP messages transmitted for this range.

Examples To display information about the range “remote”, use the command:

```
SHOW DHCP RANGE=REMOTE
```

See Also SHOW DHCP
SHOW DHCP CLIENT
SHOW DHCP POLICY

Chapter 16

Simple Network Management Protocol (SNMP)

Introduction	16-2
Network Management Framework	16-2
Structure of Management Information	16-3
Names	16-4
Instances	16-4
Syntax	16-5
Access	16-5
Status	16-5
Description	16-6
The SNMP Protocol	16-6
SNMP Messages	16-6
Polling versus Event Notification	16-8
Communities and Views	16-8
Support for SNMP	16-9
Configuration Example	16-11
Command Reference	16-12
ADD SNMP COMMUNITY	16-13
CREATE SNMP COMMUNITY	16-14
DELETE SNMP COMMUNITY	16-15
DESTROY SNMP COMMUNITY	16-16
DISABLE SNMP	16-16
DISABLE SNMP AUTHENTICATE_TRAP	16-16
DISABLE SNMP COMMUNITY	16-17
ENABLE SNMP	16-17
ENABLE SNMP AUTHENTICATE_TRAP	16-18
ENABLE SNMP COMMUNITY	16-18
SET SNMP COMMUNITY	16-19
SHOW SNMP	16-20
SHOW SNMP COMMUNITY	16-22

Introduction

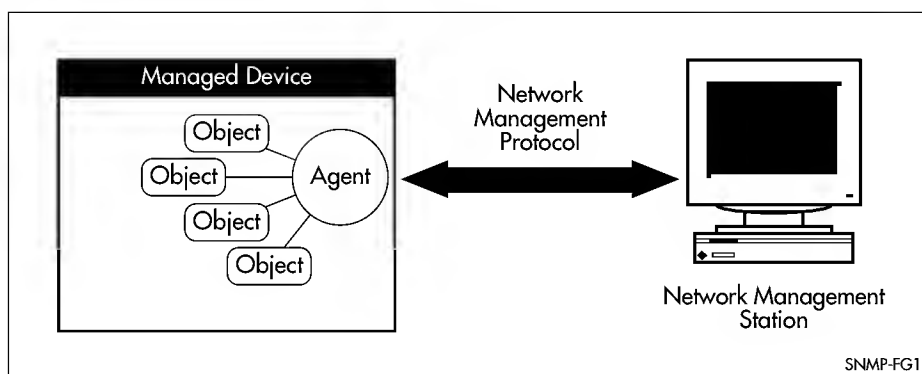
The Simple Network Management Protocol (SNMP) is the network management protocol of choice for the Internet and IP-based internetworks. This chapter describes the main features of SNMP, support for SNMP on the router, and how to configure the router's SNMP agent. See *Appendix C, SNMP MIBs* for a detailed description of all MIBs (Management Information Bases) and MIB objects supported by the router.

Network Management Framework

A network management system has three components (Figure 16-1 on page 16-2):

- One or more *managed devices*, each containing an agent which provides the management functions. A managed device may be any computing device with a network capability, for example, a host system, workstation, terminal server, printer, router, bridge, hub or repeater.
- One or more *Network Management Stations* (NMS). An NMS is a host system running a network management protocol and network management applications, enabling the user to *manage* the network.
- A *network management protocol* used by the NMS and agents to exchange information.

Figure 16-1: Components of a network management system.



The *Internet-standard Network Management Framework* is the framework used for network management in the Internet. The framework is defined by three documents:

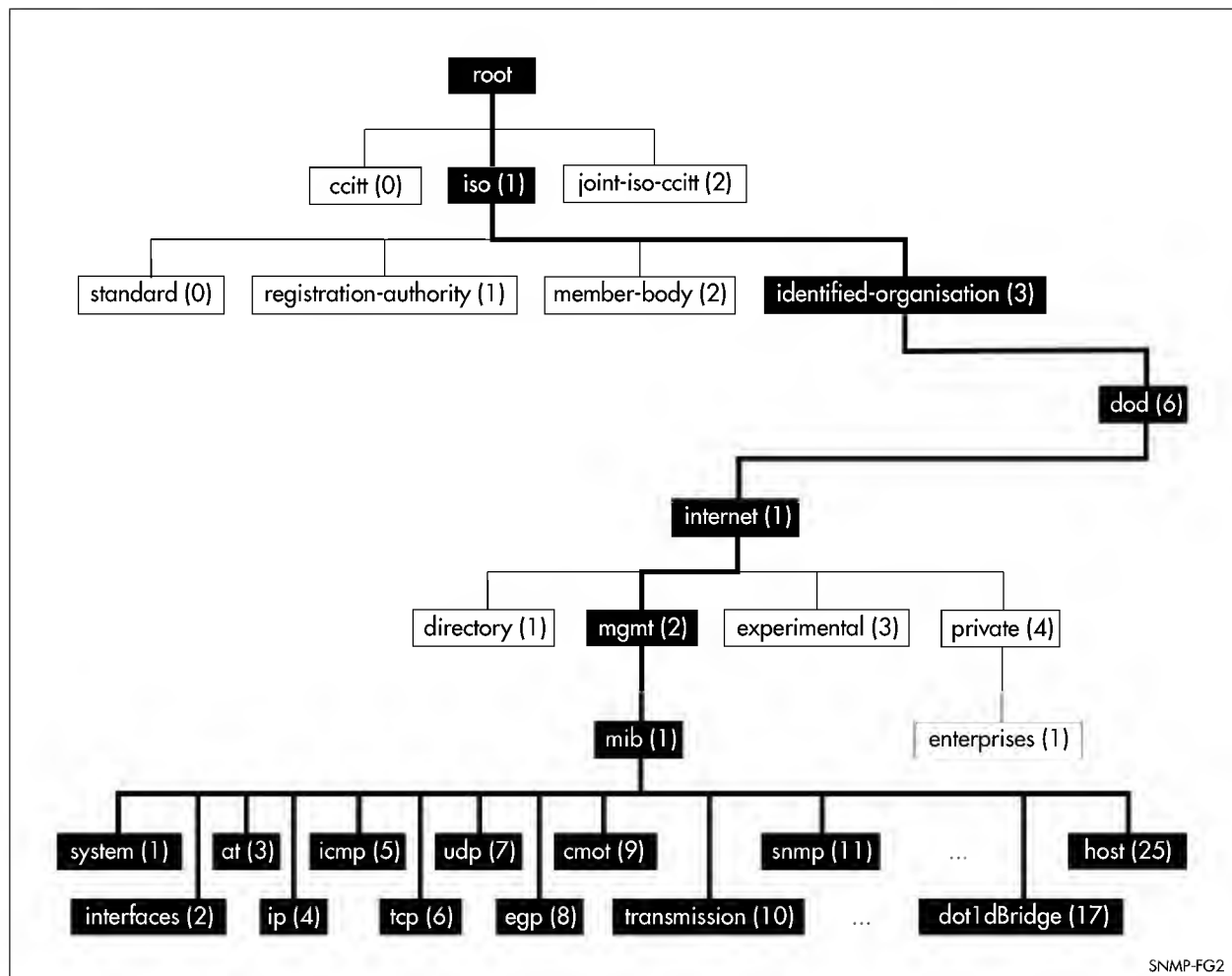
- RFC 1155, "*Structure and identification of management information for TCP/IP-based internets*" (referred to as the SMI), details the mechanisms used to describe and name the objects to be managed.
- RFC 1213, "*Management Information Base for network management of TCP/IP-based internets: MIB-II*" (referred to as MIB-II), defines the core set of managed objects for the Internet suite of protocols. The set of managed objects can be extended by adding other MIBs specific to particular protocols, interfaces or network devices.
- RFC 1157, "*A Simple Network Management Protocol (SNMP)*" (referred to as SNMP), is the protocol used for communication between management stations and managed devices.

Structure of Management Information

The SMI defines the schema for a collection of managed objects residing in a virtual store called the *Management Information Base (MIB)*. The information in a MIB includes administrative and operational configuration information, as well as counters of system events and activities.

The MIB is organised into a tree-like hierarchy. Each node in the tree has a label consisting of a non-negative integer and an optional brief textual description. The top of the MIB, as it relates to the management of Internet protocols, is summarised in Figure 16-2 on page 16-3.

Figure 16-2: The top levels of the Internet-standard Management Information Base (MIB).



Objects defined in the Internet-standard MIB (MIB-II) reside in the mib(1) sub-tree.

Managed objects are the leaf nodes in the tree. Each managed object is defined by its name, syntax, access mode, status and description.

Names

Names are used to identify managed objects, and are hierarchical in nature. An *object identifier* is a globally unique, authoritatively assigned sequence of non-negative integers which traverse the MIB tree from the root to the node containing the object.

Object identifiers may be represented in one of three forms:

- **Dotted notation** lists the integer values found by traversing the tree from the root to the node in question, separated by dots. For example:

1.3.6.1.2.1

identifies the MIB-II sub-tree, and:

1.3.6.1.2.1.1.1

identifies the *sysDescr* object in the system group of MIB-II.

- **Textual notation** lists the textual descriptions found by traversing the tree from the root to the node in question, separated by spaces and enclosed in braces. For example:

{ iso org dod 1 }

identifies the *internet* sub-tree. The name may be abbreviated to a relative form:

{ internet 1 }

identifies the first (*directory*) node of the *internet* sub-tree.

- **Combined notation** lists both the integer values and textual descriptions found by traversing the tree from the root to the node in question. The integer value is placed in parentheses after the textual description. The labels are separated by spaces and enclosed in braces. For example:

{ iso(1) org(3) dod(6) internet(1) 1 }

identifies the first (*directory*) node in the *internet* sub-tree. The name may be abbreviated to:

directory(1)

Since there is no effective limit to the magnitude of non-negative integers, and no effective limit to the depth of the tree, the MIB provides an unlimited name space.

An object is also usually assigned an *object descriptor*. The object descriptor is a unique, mnemonic, printable string intended for humans to use when discussing the MIB. Examples are *sysDescr*, *ifTable* and *ipRouteNextHop*.

Instances

Objects are just templates for data types. An actual value that can be manipulated by an NMS is an *instance* of an object. An instance is named by appending an *instance identifier* to the end of the object's object identifier. The instance identifier depends on the object's data type:

- If the object is not a column in a table, the instance identifier is 0 (zero). For example, the instance of the *sysDescr* object is:

sysDescr.0
or 1.3.6.1.2.1.1.1.0

- If the object is a column in a table, the method used to assign an instance identifier varies. Typically, the value of the index column or columns is used.

The object *ifTable* in MIB-II contains information about interfaces and is indexed by the interface number, *ifIndex*. The instance of the *ifDescr* object for the first interface is:

```
ifDescr.1
or1.3.6.1.2.1.2.2.1.2.1
```

If the index column is an IP address, the entire IP address is used as the instance identifier. The object *ipRouteTable* in MIB-II contains information about IP routes and is indexed by the destination address, *ipRouteDest*. The instance of the *ipRouteNextHop* object for the route 131.203.9.0 is:

```
ipRouteNextHop.131.203.9.0
or1.3.6.1.2.1.4.21.1.7.131.203.9.0
```

If the table has more than one index, the values of all the index columns are combined to form the instance identifier. The object *tcpConnTable* in MIB-II contains information about existing TCP connections and is indexed by the local IP address (*tcpConnLocalAddress*), the local port number (*tcpConnLocalPort*), the remote IP address (*tcpConnRemAddress*) and the remote port number (*tcpConnRemPort*) of the TCP connection. The instance of the *tcpConnState* object for the connection between 131.203.8.36,23 and 131.203.9.197,1066 is:

```
tcpConnState.131.203.8.36.23.131.203.9.197.1066
or1.3.6.1.2.1.6.13.1.1.131.203.8.36.23.131.203.9.197.1066
```

Syntax

The syntax of an object describes the abstract data structure corresponding to that object type. For example, INTEGER or OCTET STRING.

Access

The access mode of an object describes the level of access for the object (Table 16-1 on page 16-5).

Table 16-1: Access modes for MIB objects.

Access	Description
Read-only	The object's value can be read but not set.
Read-write	The object's value can be read and set.
Write-only	The object's value can be set but not read.
Not-accessible	The object's value can not be read or set.

Status

The status of an object describes the implementation requirements for the object (Table 16-2 on page 16-6).

Table 16-2: Status values for MIB objects.

Status	Description
Mandatory	Managed devices must implement the object.
Optional	Managed devices may implement the object.
Obsolete	Managed devices need no longer implement the object.
Deprecated	Managed devices should implement the object. However, the object may be deleted from the next version of the MIB. A new object with equal or superior functionality is defined.

Description

The definition of an object may include an optional textual description of the meaning and use of the object.

The SNMP Protocol

The SNMP protocol provides a mechanism for management entities, or stations, to extract information from the *Management Information Base* (MIB) of a managed device.

The normal method of accessing information in a MIB is to use a Network Management Station (NMS), typically a PC or workstation, to send commands to the managed device (in this case the router) using the SNMP protocol.

SNMP can use a number of different protocols as its underlying transport mechanism, but the most common transport protocol, and the only one supported by the router, is UDP. Therefore the IP module must be enabled and properly configured in order to use SNMP. SNMP *trap* messages are sent to UDP port 162; all other SNMP messages are sent to UDP port 161. The router's SNMP agent accepts SNMP messages up to the maximum UDP length the router can receive.



Other transport mappings have been defined (e.g. OSI [RFC 1418], AppleTalk [RFC 1419] and IPX [RFC 1420]), but the standard transport mapping for the Internet (and the one used by the router) is UDP. The IP module must be enabled and configured correctly. See Chapter 6, Internet Protocol (IP) for detailed descriptions of the commands required to enable and configure IP.

SNMP Messages

The SNMP protocol is termed *simple* because it has only five operations, or messages—*get*, *get-next*, *get-response*, *set*, and *trap* (Table 16-4 on page 16-7). The replies from the managed device are processed by the NMS and generally used to provide a graphical representation of the state of the network. The two major SNMP operations available to a management station for interacting with a client are the *get* and *set* operations. The SNMP *set* operator can lead to security breaches, since SNMP is not inherently very secure. Care must be taken in the choice and safe-guarding of community names, which are effectively passwords for SNMP. See *Appendix C, SNMP MIBs* for a description of the router's implementation of each MIB object with read-write access.

Figure 16-3 on page 16-7 shows the format of an SNMP message. The function of the fields are described in Table 16-3 on page 16-7. There are five different SNMP PDUs (Table 16-4 on page 16-7) and seven generic traps (Table 16-5 on page 16-7).

Figure 16-3: Format of an SNMP message.

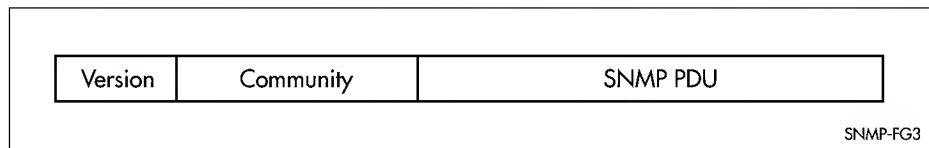


Table 16-3: Fields in an SNMP message.

Field	Function
Version	The version of the SNMP protocol. The value is version-1 (0) for the SNMP protocol as defined in RFC 1157.
Community	The name of an SNMP community, for authentication purposes.
SNMP PDU	An SNMP Protocol Data Unit (PDU)

Table 16-4: SNMP PDUs.

PDU	Function
get-request	Sent by an NMS to an agent, to retrieve the value of an object.
get-next-request	Sent by an NMS to an agent, to retrieve the value of the next object in the sub-tree. A sub-tree is traversed by issuing a get-request PDU followed by successive get-next-request PDUs.
set-request	Sent by an NMS to an agent, to manipulate the value of an object.
get-response	Sent by an agent to an NMS in response to a get-request, get-next-request or set-request PDU.
trap	Sent by an agent to an NMS, to notify the NMS of a extraordinary event.

Table 16-5: Generic SNMP traps.

Value	Meaning
coldStart (0)	The agent is re-initialising itself. Objects may be altered.
warmStart (1)	The agent is re-initialising itself. Objects will not be altered.
linkDown (2)	An interface has changed state from "Up" to "Down".
linkUp (3)	An interface has changed state from "Down" to "Up".
authenticationFailure (4)	An SNMP message has been received with an invalid community name.
egpNeighborLoss (5)	An EGP peer has transitioned to state "Down".
enterpriseSpecific (6)	Some other enterprise-specific trap.

Polling versus Event Notification

SNMP employs a *polling* paradigm. A Network Management Station (NMS) polls the managed device for information as and when it is required, by sending *get-request* and/or *get-next-request* PDUs to the managed device. The managed device responds by returning the requested information in a *get-response* PDU. The NMS may manipulate objects in the managed device by sending a *set-request* PDU to the managed device.

The only time that a managed device may initiate an exchange of information is the special case of a *trap* PDU. A managed device may generate a limited set of traps to notify the NMS of critical events that may affect the ability of the NMS to communicate with the managed device or other managed devices on the network, and therefore to “manage” the network. Such events include the restarting or re-initialisation of a device, a change in the status of a network link (up or down), or an authentication failure.

Communities and Views

A *community* is a relationship between an NMS and an agent. The community name is used like a password for a trivial authentication scheme.

An SNMP MIB *view* is a arbitrary subset of objects in the MIB. Objects in the view may be from any part of the object name space, and not necessarily the same sub-tree. An *SNMP community profile* is the pairing of an *SNMP access mode* (*read-only* or *read-write*) with the access mode defined by the MIB for each object in the view. For each object in the view, the community profile defines the operations that can be performed on the object (Table 16-6 on page 16-8).

Table 16-6: Community profiles for objects in a MIB view.

SNMP Access Mode	Object Access Defined by MIB			
	Read-Only	Read-Write	Write-Only	Not Accessible
Read-Only	get, get-next, trap	get, get-next, trap	None	None
Read-Write	get, get-next, trap	get, get-next, set, trap	get, get-next, set, trap(*)	None

A pairing of an SNMP community and an SNMP community profile determines the level of access that the agent affords to an NMS that is a member of the specified community. When an agent receives an SNMP message it checks the community name encoded in the message. If the agent knows the community name, the message is deemed to be an authentic SNMP message and the sending SNMP entity is accepted as a member of the community. The community profile associated with the community name then determines the sender’s view of the MIB and the operations that can be performed on objects in the view.

Support for SNMP

The router's implementation of SNMP is based on RFC 1157 "A Simple Network Management Protocol (SNMP)", and RFC 1812, "Requirements for IP Version 4 Routers".

The router's SNMP agent can be enabled or disabled using the commands:

```
ENABLE SNMP
DISABLE SNMP
```

When the SNMP agent is disabled, the agent will not respond to any SNMP request messages. The agent is disabled by default. The current state and configuration of the SNMP agent can be displayed using the command:

```
SHOW SNMP
```

An SNMP *community* is a pairing of an SNMP agent with a set of SNMP application entities.

SNMP communities are the main configuration item in the router's implementation of SNMP, and are defined in terms of a list of IP addresses which define the SNMP application entities (trap hosts and management stations) in the community. An SNMP community is created using the command:

```
CREATE SNMP COMMUNITY=name [ACCESS={READ|WRITE}]
[TRAPHOST=ipadd] [MANAGER=ipadd]
[OPEN={ON|OFF|YES|NO|TRUE|FALSE}]
```

which defines the name of the community (e.g. "public"), and specifies the IP address of a trap host and/or a management station. A community can be modified using the command:

```
SET SNMP COMMUNITY=name [ACCESS={READ|WRITE}]
[OPEN={ON|OFF|YES|NO|TRUE|FALSE}]
```



Community names act as passwords and provide only trivial authentication. Any SNMP application entity that knows a community name can read the value of any instance of any object in the MIB implemented in the router. Any SNMP application entity that knows the name of a community with write access can change the value of any instance of any object in the MIB implemented in the router, possibly affecting the operation of the router. For this reason, care must be taken with the security of community names.

An SNMP community is destroyed using the command:

```
DESTROY SNMP COMMUNITY=name
```

Additional trap hosts and management stations can be added to or removed from a community using the commands:

```
ADD SNMP COMMUNITY=name [TRAPHOST=ipadd] [MANAGER=ipadd]
DELETE SNMP COMMUNITY=name [TRAPHOST=ipadd] [MANAGER=ipadd]
```

When a trap is generated by the SNMP agent it is forwarded to all the trap hosts assigned to the community. The community name and manager addresses are used to provide trivial authentication. An incoming SNMP message is deemed authentic if it contains a valid community name and originated from an IP address defined as a management station for that community.

An SNMP community, or the generation of traps by the community, can be temporarily enabled or disabled using the commands:

```
DISABLE SNMP COMMUNITY=name [TRAP]
ENABLE SNMP COMMUNITY=name [TRAP]
```

When a community is disabled, the SNMP agent behaves as if the community does not exist, and will generate authentication failure traps for messages directed to the disabled community. Information about the configuration of SNMP communities can be displayed using the command:

```
SHOW SNMP COMMUNITY=name
```



The SNMP agent does not support a default community called “public” with read-only access, traps disabled and open access as mandated in RFC 1812, as this is a security hole open for users who wish to use the router with minimal modification to the default configuration. The default configuration of the router has no defined communities. Communities must be explicitly created. The defaults for other parameters such as the open access flag and the trap enabled flag also follow the principle of security first, access second.

SNMP *authentication* is a mechanism whereby an SNMP message is declared to be authentic, that is from an SNMP application entity actually in the community to which the message purports to belong. The mechanism may be trivial or secure. The only form of SNMP authentication implemented by the router’s SNMP agent is trivial authentication. The authentication failure trap may be generated as a result of the failure to authenticate an SNMP message. The generation of authentication failure traps may be enabled or disabled using the commands:

```
ENABLE SNMP AUTHENTICATE_TRAP
DISABLE SNMP AUTHENTICATE_TRAP
```

Link up/down traps can be enabled or disabled on a per-interface basis, using the commands:

```
ENABLE INTERFACE={ifIndex|interface|DYNAMIC} LINKTRAP
DISABLE INTERFACE={ifIndex|interface|DYNAMIC} LINKTRAP
```

where *ifIndex* is the value of *ifIndex* for the interface in the Interface Table and *interface* is the name of the interface. If link traps are enabled, when an interface changes to or from the “Down” state an SNMP trap is sent to any defined trap hosts. Link traps are disabled by default on the router. The current settings for link traps can be displayed using the command:

```
SHOW INTERFACE={ifIndex|interface}
```

The maximum number of link traps generated per minute can be set for each static interface or for all dynamic interfaces, using the command:

```
SET INTERFACE={ifIndex|interface|DYNAMIC} TRAPLIMIT=1..60
```

See *Chapter 2, Interfaces* for a detailed description of the commands for configuring and monitoring link up/down traps.

Router interfaces can be enabled or disabled via SNMP by setting the *ifAdminStatus* object in the *ifTable* of MIB-II MIB to ‘Up(1)’ or ‘Down(2)’ for the corresponding *ifIndex*. If it is not possible to change the status of a particular interface the router will return an SNMP error message.

The router’s implementation of the *ifOperStatus* object in the *ifTable* of MIB-II MIB supports two additional values—“Unknown(4)” and “Dormant(5)” (e.g. an inactive dial-on-demand interface).



An unauthorised person, with knowledge of the appropriate SNMP community name, could bring an interface up or down. Community names act as passwords for the SNMP protocol. Care should be taken when creating an SNMP community with write access to select a secure community name and to ensure that this name is known only to authorised personnel.

An SNMP MIB *view* is a subset of objects in the MIB that pertain to a particular network element. For example, the MIB view of a hub would be the objects relevant to management of the hub, and would not include IP routing table objects, for example. The router's SNMP agent does not allow the construction of MIB views. The router supports all relevant objects from all MIBs that it implements.



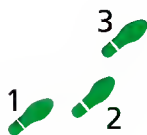
The router's standard SET and SHOW commands can also be used to access objects in the MIBs supported by the router.

Configuration Example

The following example illustrates the steps required to configure the router's SNMP agent. In this example, two network management stations have been set up on a large network. The central NMS (IP address 192.168.11.5) will be used to both monitor devices on the network and use SNMP *set* messages to manage the devices on the network. Trap messages will be sent to this management station. The regional network management station (IP addresses 192.168.16.1) will be used just to monitor devices on the network using SNMP *get* messages. Link traps will be enabled for all interfaces on this particular router.



The IP module must be enabled and correctly configured in order to access the SNMP agent in the router, since the IP module handles the UDP datagrams used to transport SNMP messages. See Chapter 6, Internet Protocol (IP) for a detailed description of the commands required to enable and configure IP.



To configure SNMP:

1. Enable the SNMP agent.

Enable the SNMP agent and enable the generation of authenticate failure traps to monitor unauthorised SNMP access.

```
ENABLE SNMP  
ENABLE SNMP AUTHENTICATE_TRAP
```

2. Create a community with write access for the central NMS.

Create a community called "private", with write access for use only by the central network management station at 192.168.11.5. All traps will be sent to this NMS.

```
CREATE SNMP COMMUNITY=private ACCESS=WRITE  
TRAPHOST=192.168.11.5 MANAGER=192.168.11.5 OPEN=NO
```



Do not use the name “private” in a real network—it’s too obvious! Use a different name! Community names act as passwords and provide only trivial authentication. Any SNMP application entity that knows a community name can read the value of any instance of any object in the MIB implemented in the router. Any SNMP application entity that knows the name of a community with write access can change the value of any instance of any object in the MIB implemented in the router, possibly affecting the operation of the router. For this reason, care must be taken with the security of community names.

3. Create a community with read-only access for the regional NMS.

Create a community called public, with read-only access for use by the regional network management station at 192.168.16.1.

```
CREATE SNMP COMMUNITY=public ACCESS=READ  
MANAGER=192.168.16.1 OPEN=NO
```

4. Enable link traps.

This router has static interfaces ppp0 and x25t0. Additional dynamic interfaces may be created and destroyed as the result of ISDN calls. Enable link traps for these interfaces, and set a limit of 30 traps per minute for dynamic interfaces.

```
ENABLE INTERFACE=ppp0 LINKTRAP  
ENABLE INTERFACE=x25t0 LINKTRAP  
ENABLE INTERFACE=DYNAMIC LINKTRAP  
SET INTERFACE=DYNAMIC TRAPLIMIT=30
```

5. Check the configuration.

Check that the current configuration of the SNMP communities matches the desired configuration:

```
SHOW SNMP  
SHOW SNMP COMMUNITY
```

Check that the interface link up/down traps have been correctly configured:

```
SHOW INTERFACE=ppp0  
SHOW INTERFACE=x25t0  
SHOW INTERFACE
```

Command Reference

This section describes the commands available on the router to configure and manage the SNMP agent. The IP module must be enabled and correctly configured in order to access the SNMP agent in the router, since the IP module handles the UDP datagrams used to transport SNMP messages. See *Chapter 6, Internet Protocol (IP)* for a detailed description of the commands required to enable and configure IP.

See “Conventions” on page xxxv of *Preface* in the front of this manual for details of the conventions used to describe command syntax. See *Appendix A, Messages* for a complete list of error messages and their meanings.

ADD SNMP COMMUNITY

Syntax `ADD SNMP COMMUNITY=name [TRAPHOST=ipadd] [MANAGER=ipadd]`

where:

- *name* is a character string, 1 to 15 characters in length. Valid characters are uppercase letter (A–Z), lowercase letters (a–z) and digits (0–9). *name* is case-sensitive, that is “Public” is a different name from “public”.
- *ipadd* is an IP address in dotted decimal notation.

Description This command adds a trap host or a management station to the specified SNMP community.

The COMMUNITY parameter specifies the SNMP community. The community must already exist on the router.

The TRAPHOST parameter specifies a trap host for the SNMP community. This is the IP address of a device to which traps generated by the router will be sent. A community may have more than one trap host.

The MANAGER parameter specifies a management station for this SNMP community. This is the IP address of a device from which SNMP requests with the community name will be deemed to be authentic. A community may have more than one management station.

Examples To add the host 192.168.1.1 as both a trap host and a management station to the existing SNMP community “Administration”, use the command:

```
ADD SNMP COMMUNITY=Administration TRAPHOST=192.168.1.1
MANAGER=192.168.1.1
```

See Also CREATE SNMP COMMUNITY
DELETE SNMP COMMUNITY
DESTROY SNMP COMMUNITY
DISABLE SNMP COMMUNITY
ENABLE SNMP COMMUNITY
SET SNMP COMMUNITY
SHOW SNMP COMMUNITY

CREATE SNMP COMMUNITY

Syntax `CREATE SNMP COMMUNITY=name [ACCESS={READ|WRITE}]`
`[TRAPHOST=ipadd] [MANAGER=ipadd]`
`[OPEN={ON|OFF|YES|NO|TRUE|FALSE}]`

where:

- *name* is a character string, 1 to 15 characters in length. Valid characters are uppercase letter (A–Z), lowercase letters (a–z) and digits (0–9). *name* is case-sensitive, that is “Public” is a different name from “public”.
- *ipadd* is an IP address in dotted decimal notation.

Description This command creates an SNMP community, optionally setting the access mode for the community and defining a trap host and manager.

The COMMUNITY parameter specifies the name of the community. The community name is used to reference the SNMP community in all other SNMP commands. A community with the specified name must not already exist in the router.

The ACCESS parameter specifies the access mode for this community. If READ is specified, management stations in this community can only read MIB variables from the router, that is perform SNMP *get* or *get-next* operations. If WRITE is specified, management stations in this community can read and write MIB variables, that is perform SNMP *set*, *get* and *get-next* operations. The default is READ.

The TRAPHOST parameter specifies a trap host for the SNMP community. This is the IP address of a device to which traps generated by the router will be sent. A community may have more than one trap host, but only one can be specified when the community is created. If the parameter is not specified, the community will have no defined trap host.

The MANAGER parameter specifies a management station for this SNMP community. This is the IP address of a device from which SNMP requests with the community name will be deemed to be authentic. A community can have more than one management station, but only one can be specified when the community is created. If the parameter is not specified, the community will have no defined management station.

The OPEN parameter allows access to this community by any management station, and overrides the management stations defined with the MANAGER parameter. The default is OFF.

Additional trap hosts and management stations can be defined for a the community using the ADD SNMP COMMUNITY command on page 16-13.



For security reasons this command will only be accepted if the user has SECURITY OFFICER privilege.

Examples To create an SNMP community called “public” with read only access to all MIB variables from any management station, use the command:

```
SET SNMP COMMUNITY=public OPEN=ON
```


See Also ADD SNMP COMMUNITY
 DELETE SNMP COMMUNITY
 DESTROY SNMP COMMUNITY
 DISABLE SNMP
 DISABLE SNMP COMMUNITY
 ENABLE SNMP
 ENABLE SNMP COMMUNITY
 SET SNMP COMMUNITY
 SHOW SNMP COMMUNITY

DELETE SNMP COMMUNITY

Syntax DELETE SNMP COMMUNITY=*name* [TRAPHOST=*ipadd*]
 [MANAGER=*ipadd*]

where:

- *name* is a character string, 1 to 15 characters in length. Valid characters are uppercase letter (A–Z), lowercase letters (a–z) and digits (0–9). *name* is case-sensitive, that is “Public” is a different name from “public”.
- *ipadd* is an IP address in dotted decimal notation.

Description This command deletes a trap host or management station from the specified SNMP community.

The COMMUNITY parameter specifies the SNMP community. The community must already exist on the router.

The TRAPHOST parameter specifies a trap host for the SNMP community. This is the IP address of a device to which traps generated by the router are currently sent.

The MANAGER parameter specifies a single management station for this SNMP community. This is the IP address of a device from which SNMP requests received with the community name are deemed to be authentic.

Examples To delete the host 192.168.1.1 as a trap host from the community “Administration”, use the command:

```
DELETE SNMP COMMUNITY=Administration TRAPHOST=192.168.1.1
```

See Also ADD SNMP COMMUNITY
 CREATE SNMP COMMUNITY
 DESTROY SNMP COMMUNITY
 DISABLE SNMP COMMUNITY
 ENABLE SNMP COMMUNITY
 SET SNMP COMMUNITY
 SHOW SNMP COMMUNITY

DESTROY SNMP COMMUNITY

Syntax DESTROY SNMP COMMUNITY=*name*

where:

- *name* is a character string, 1 to 15 characters in length. Valid characters are uppercase letter (A–Z), lowercase letters (a–z) and digits (0–9). *name* is case-sensitive, that is “Public” is a different name from “public”.

Description This command destroys an existing SNMP community.

The COMMUNITY parameter specifies the SNMP community. The community must already exist on the router.

See Also ADD SNMP COMMUNITY
CREATE SNMP COMMUNITY
DISABLE SNMP COMMUNITY
ENABLE SNMP COMMUNITY
SET SNMP COMMUNITY
SHOW SNMP COMMUNITY

DISABLE SNMP

Syntax DISABLE SNMP

Description This command disables the router’s SNMP agent. SNMP packets sent to the router will be treated as unknown protocol packets by the underlying transport layer (UDP) and traps will not be generated by the router.

See Also DISABLE SNMP COMMUNITY
ENABLE SNMP
ENABLE SNMP COMMUNITY
SHOW SNMP
SHOW SNMP COMMUNITY

DISABLE SNMP AUTHENTICATE_TRAP

Syntax DISABLE SNMP AUTHENTICATE_TRAP

Description This command disables the generation of authentication failure traps by the SNMP agent whenever an SNMP authentication failure occurs.

See Also DISABLE SNMP
ENABLE SNMP
ENABLE SNMP AUTHENTICATE_TRAP
SHOW SNMP

DISABLE SNMP COMMUNITY

Syntax `DISABLE SNMP COMMUNITY=name [TRAP]`

where:

- *name* is a character string, 1 to 15 characters in length. Valid characters are uppercase letter (A–Z), lowercase letters (a–z) and digits (0–9). *name* is case-sensitive, that is “Public” is a different name from “public”.

Description This command disables a particular SNMP community or disables the generation of trap messages for the community.

The COMMUNITY parameter specifies the SNMP community. The community must already exist on the router. When a community is disabled, packets for the community are processed as if the community does not exist and traps will not be generated for the community. The SNMP agent will generate an authentication error if a packet is received for a disabled community.

The TRAP parameter specifies that only traps for the community should be disabled, not the entire operation of the community. Trap messages will not be sent to the community’s trap host(s), but all other SNMP operations will proceed as normal.

See Also DISABLE SNMP
ENABLE SNMP
ENABLE SNMP COMMUNITY
SHOW SNMP
SHOW SNMP COMMUNITY

ENABLE SNMP

Syntax `ENABLE SNMP`

Description This command enables the router’s SNMP agent. The SNMP agent will receive and process SNMP packets sent to the router and generate traps.

By default, the SNMP agent is disabled. This command is required to enable SNMP to operate at boot.

See Also DISABLE SNMP
DISABLE SNMP COMMUNITY
ENABLE SNMP COMMUNITY
SHOW SNMP
SHOW SNMP COMMUNITY

ENABLE SNMP AUTHENTICATE_TRAP

Syntax ENABLE SNMP AUTHENTICATE_TRAP

Description This command enables the generation of authentication failure traps by the SNMP agent whenever an SNMP authentication failure occurs.

By default, the generation of authentication traps is disabled. This command is required to enable SNMP authentication failure traps at boot.

See Also DISABLE SNMP
DISABLE SNMP AUTHENTICATE_TRAP
ENABLE SNMP
SHOW SNMP

ENABLE SNMP COMMUNITY

Syntax ENABLE SNMP COMMUNITY=*name* [TRAP]

where:

- *name* is a character string, 1 to 15 characters in length. Valid characters are uppercase letter (A–Z), lowercase letters (a–z) and digits (0–9). *name* is case-sensitive, that is “Public” is a different name from “public”.

Description This command enables a particular SNMP community or enables the generation of trap messages for the community.

The COMMUNITY parameter specifies the SNMP community. The community must already exist on the router. When a community is enabled, the SNMP agent processes SNMP packets for the community and generates traps to trap hosts in the community, if traps are also enabled. SNMP communities are enabled when they are created, but traps are not enabled for the community.

The TRAP parameter specifies that only traps for the community should be enabled, not the entire operation of the community. Trap messages will be sent to the community's trap host(s).



For security reasons this command will only be accepted if the user has SECURITY OFFICER privilege.

Examples To create an SNMP community and enable traps on it, use the following commands:

```
CREATE SNMP COMMUNITY=private TRAPHOST=192.168.1.1
MANAGER=192.168.1.1
ENABLE SNMP COMMUNITY=private TRAP
```

See Also DISABLE SNMP
DISABLE SNMP COMMUNITY
ENABLE SNMP
SHOW SNMP
SHOW SNMP COMMUNITY

SET SNMP COMMUNITY

Syntax SET SNMP COMMUNITY=*name* [ACCESS={READ|WRITE}]
[OPEN={ON|OFF|YES|NO|TRUE|FALSE}]

where:

- *name* is a character string, 1 to 15 characters in length. Valid characters are uppercase letter (A–Z), lowercase letters (a–z) and digits (0–9). *name* is case-sensitive, that is “Public” is a different name from “public”.

Description This command modifies the access mode and open access configuration for the specified SNMP community.

The COMMUNITY parameter specifies the name of the community. A community with the specified name must already exist in the router.

The ACCESS parameter specifies the access mode for this community. If READ is specified, management stations in this community can only read MIB variables from the router, that is perform SNMP *get* or *get-next* operations. If WRITE is specified, management stations in this community can read and write MIB variables, that is perform SNMP *set*, *get* and *get-next* operations. The default is READ.

The OPEN parameter allows access to this community by any management station, and overrides the management stations defined with the MANAGER parameter. The default is OFF.



For security reasons this command will only be accepted if the user has SECURITY OFFICER privilege.

Examples To disable access from any management station for an SNMP community called “public”, use the command:

```
SET SNMP COMMUNITY=public OPEN=OFF
```

See Also CREATE SNMP COMMUNITY
DESTROY SNMP COMMUNITY
SHOW SNMP COMMUNITY

SHOW SNMP

Syntax SHOW SNMP

Description This command displays information about the router's SNMP agent (Figure 16-4 on page 16-20, Table 16-7 on page 16-20).

Figure 16-4: Example output from the SHOW SNMP command.

```
SNMP configuration:
  Status ..... Enabled
  Authentication failure traps .... Enabled
  Community ..... public
    Access ..... read-only
    Status ..... Enabled
    Traps ..... Enabled
    Open access ..... Yes
  Community ..... Administration
    Access ..... read-write
    Status ..... Disabled
    Traps ..... Disabled
    Open access ..... No

SNMP counters:
  inPkts ..... 0          outPkts ..... 0
  inBadVersions ..... 0    outTooBigs ..... 0
  inBadCommunityNames ..... 0    outNoSuchNames ..... 0
  inBadCommunityUses ..... 0    outBadValues ..... 0
  inASNParseErrs ..... 0        outGenErrs ..... 0
  inTooBigs ..... 0            outGetRequests ..... 0
  inNoSuchNames ..... 0        outGetNexts ..... 0
  inBadValues ..... 0          outSetRequests ..... 0
  inReadOnly ..... 0           outGetResponses ..... 0
  inGenErrs ..... 0            outTraps ..... 0
  inTotalReqVars ..... 0
  inTotalSetVars ..... 0
  inGetRequests ..... 0
  inGetNexts ..... 0
  inSetRequests ..... 0
  inGetResponses ..... 0
  inTraps ..... 0
```

Table 16-7: Parameters displayed in the output of the SHOW SNMP command.

Parameter	Meaning
Status	The status of the SNMP agent or the specified community; one of "Enabled" or "Disabled".
Authentication failure traps	Whether or not the SNMP agent will generate a trap on an authentication failure for an incoming SNMP packet; one of "Enabled" or "Disabled".
Community	The name of an SNMP community on the router.
Access	The access rights for the SNMP community; one of "read-only" or "read-write".
Status	The status of the community; one of "Enabled" or "Disabled".

Table 16-7: Parameters displayed in the output of the SHOW SNMP command. (Continued)

Parameter	Meaning
Traps	Whether or not the community will generate traps; one of "Enabled" or "Disabled".
Open access	Whether or not the SNMP community is open to access from all IP addresses; one of "Yes" or "No".
inPkts	The number of SNMP packets received by the router.
inBadVersions	The number of SNMP packets with a bad version field received by the router.
inBadCommunityNames	The total number of SNMP PDUs delivered to the SNMP agent that used an unknown SNMP community name.
inBadCommunityUses	The total number of SNMP PDUs delivered to the SNMP agent that represented an SNMP operation not allowed by the SNMP community name in the PDU.
inASNParseErrs	The total number of ASN.1 parsing errors, either in encoding or syntax, encountered by the SNMP agent when decoding received SNMP PDUs.
inTooBigs	The total number of valid SNMP PDUs delivered to the SNMP agent for which the value of the errorStatus component was tooBig.
inNoSuchNames	The number of SNMP packets received with an error status of nosuchname.
inBadValues	The number of SNMP packets received with an error status of badvalue.
inReadOnlys	The number of SNMP packets received with an error status of readonly.
inGenErrs	The number of SNMP packets received with an error status of generr.
inTotalReqVars	The total number of SNMP MIB objects requested.
inTotalSetVars	The total number of SNMP MIB objects which were changed.
inGetRequests	The number of SNMP Get Request packets received by the router.
inGetNexts	The number of SNMP Get Next Request packets received by the router.
inSetRequests	The number of SNMP Set Request packets received by the router.
inGetResponses	The number of SNMP Get Response packets received by the router.
inTraps	The number of SNMP trap message packets received by the router.
outPkts	The number of SNMP packets transmitted by the router.
outTooBigs	The number of SNMP packets transmitted with an error status of toobig.
outNoSuchNames	The number of SNMP packets transmitted with an error status of nosuchname.
outBadValues	The number of SNMP packets transmitted with an error status of badvalue.

Table 16-7: Parameters displayed in the output of the SHOW SNMP command. (Continued)

Parameter	Meaning
outGenErrs	The number of SNMP packets transmitted with an error status of generror.
outGetRequests	The number of SNMP Get Request response packets transmitted by the router.
outGetNexts	The number of Get Next response packets transmitted by the router.
outSetRequests	The number of Set Request packets transmitted by the router.
outGetResponses	The number of SNMP Get response packets transmitted.
outTraps	The number of SNMP Traps transmitted by the router.

See Also SHOW SNMP COMMUNITY

SHOW SNMP COMMUNITY

Syntax SHOW SNMP COMMUNITY=*name*

where:

- *name* is a character string, 1 to 15 characters in length. Valid characters are uppercase letter (A–Z), lowercase letters (a–z) and digits (0–9). *name* is case-sensitive, that is “Public” is a different name from “public”.

Description This command displays information about a single SNMP community (Figure 16-5 on page 16-22, Table 16-8 on page 16-23).

The COMMUNITY parameter specifies the name of the community. A community with the specified name must already exist in the router.

Figure 16-5: Example output from the SHOW SNMP COMMUNITY command.

```
SNMP community information:
  Name ..... public
  Access ..... read-only
  Status ..... Enabled
  Traps ..... Enabled
  Open access ..... Yes
  Manager ..... 192.168.1.1
  Manager ..... 192.168.5.3
  Trap host ..... 192.168.1.1
  Trap host ..... 192.168.6.23
```


Table 16-8: Parameters displayed in the output of the SHOW SNMP COMMUNITY command.

Parameter	Meaning
Name	The name of the community. This identifies the community and appears in SNMP messages for this community.
Access	The access rights for the SNMP community; one of "read-only" or "read-write".
Status	The status of the community; one of "Enabled" or "Disabled".
Traps	Whether or not the community generates trap messages; one of "Enabled" or "Disabled".
Open access	Whether or not the community is open to access from all IP addresses; one of "Yes" or "No".
Manager	The IP address of a management station that can access this router using this community.
Trap host	The IP address of a trap host to which traps for this community will be sent.

See Also SHOW SNMP

Chapter 17

Firewall

Introduction	17-2
Policies	17-3
Rules	17-4
NAT	17-6
Monitoring Firewall Activity	17-6
Debugging	17-6
Logging	17-6
Configuration Examples	17-8
Minimum Configuration for a Small Office	17-8
A Firewall with an ISP-assigned Internet Address	17-9
A Firewall with a Single Global Internet Address	17-10
Allowing Access to a WWW Server	17-10
Command Reference	17-11
ADD FIREWALL POLICY INTERFACE	17-11
ADD FIREWALL POLICY NAT	17-12
ADD FIREWALL POLICY RULE	17-14
CREATE FIREWALL POLICY	17-16
DELETE FIREWALL POLICY INTERFACE	17-17
DELETE FIREWALL POLICY NAT	17-18
DELETE FIREWALL POLICY RULE	17-19
DELETE FIREWALL SESSION	17-19
DESTROY FIREWALL POLICY	17-20
DISABLE FIREWALL	17-20
DISABLE FIREWALL POLICY	17-21
ENABLE FIREWALL	17-22
ENABLE FIREWALL POLICY	17-22
SET FIREWALL POLICY RULE	17-24
SHOW FIREWALL	17-25
SHOW FIREWALL POLICY	17-26
SHOW FIREWALL SESSION	17-32

Introduction

This chapter describes the router's built-in firewall facility, and how to configure and monitor the firewall.

The Internet is a network which allows access to vast amounts of information and potential customers. However, the Internet is not controlled and certain individuals use it destructively. These individuals attack other user's computer systems for entertainment and/or profit.

A firewall is a security device designed to allow safe access to the Internet by enforcing a set of access rules between the various interfaces of the product. Typically a firewall has two interfaces—one interface is attached to the public network (e.g. the Internet), and the other interface is attached to an internal private network (intranet) which requires protection. The firewall prevents unrestricted access to the private network and protects the computer systems behind the firewall from attack.

Because a firewall provides a single link between the private network and the public network, a firewall is also uniquely positioned to provide a single point where all traffic in to and out of the private network can be logged and monitored. This information is useful for providing a security audit trail.

Currently two main firewall technologies are recognised:

■ Application Gateway

This is the traditional approach used to build a firewall, where every connection between two networks is made via an application program (called a *proxy*) specific for that protocol. A session from the private network is terminated by the proxy, which then creates another separate session to the end destination. Typically, a proxy is designed with a detailed knowledge of how the protocol works and what is and is not allowed. This approach is very CPU intensive and very restrictive. Only protocols that have specific proxies configured are allowed through the firewall; all other traffic is rejected. In practice most third-party proxies are transparent proxies which pass all traffic between the two sessions without regard to the data.

■ Stateful Inspection

A more recent approach to firewall design uses a method called "*stateful inspection*". Stateful inspection is also referred to as *dynamic packet filtering* or *context-based access control* (CBAC). In this technology, an inspection module understands data in packets from the network layer (IP headers) up to the application layer. The inspection module checks every packet passing through the firewall and makes access decisions based on the source, destination and service requested. The term *stateful* refers to the firewall's ability to remember the status of a flow, for example, whether a packet from the public Internet is returning traffic for a flow originated from the private intranet. The TCP state of TCP flows is also monitored, allowing inappropriate traffic to be discarded. The benefit of this approach is that stateful inspection firewalls are generally faster, less demanding on hardware and more adaptive to new Internet applications.

The router's firewall implementation has the following features:

- Dynamic packet filtering (stateful inspection) technology.
- Application of dynamic filtering to traffic flows, using the base rule that all access from the outside (i.e. public interfaces) is denied unless specifically

permitted and all access from the inside (i.e. private interfaces) is allowed unless specifically denied.

- The firewall will open only the required ports for the duration of a user session. Configuration commands are required to allow access to internal hosts from a public interface.
- The firewall intercepts all TCP connections and completes the connection. This feature better tracks and defends against denial of service attacks by depletion of TCP slots. Any further out-of-sequence TCP frames are dropped.
- The firewall can be configured to limit internal access to the public network based on a policy setting.
- The generation of unreachable ICMP messages can be enabled or disabled.
- All firewall events can be selectively logged to the Logging Facility.
- The firewall supports protocols such as FTP, RealAudio from Progressive Networks, Streamworks from Xing Technologies, CuSeeMe from White Pines, VDOLive from VDOnet, QuickTime streaming video from Apple Computer, Microsoft NetShow, NETBIOS, GRE, OSPF, PPTP and RSVP.
- The firewall detects and logs a range of denial of service attacks including SYN and FIN flooding, Ping of death (illegal ping packet sizes, or an excessive number of ICMP messages), Smurf attacks (packets with an IP address of the private network and typically a broadcast address) and port scans.
- An accounting facility records, via the Logging Facility, the traffic flow for an individual session.

Policies

The first step in deploying a firewall is to determine exactly what traffic should be allowed and what traffic should be denied. This is called the *security policy*. The security policy will contain rules that specify the particular types of traffic that are or are not allowed to pass through the firewall. The configuration of the firewall is based around this concept of a security policy.

The firewall is enabled or disabled using the commands:

```
ENABLE FIREWALL
DISABLE FIREWALL
```

The current status and configuration summary can be displayed using the command:

```
SHOW FIREWALL
```

A policy is created or destroyed using the commands:

```
CREATE FIREWALL POLICY=name
DESTROY FIREWALL POLICY=name
```

The firewall will not become active until at least one public and one private interface have been assigned to the policy. A public interface is an interface attached to a public network such as the Internet. A private interface is an interface attached to a private network, such as a company intranet, behind the firewall. The basic function of a firewall is to control the forwarding of traffic

between the public interface and the private interface. Interfaces are added to or removed from a policy using the commands:

```
ADD FIREWALL POLICY=name INTERFACE=interface TYPE={PUBLIC|
PRIVATE} [METHOD={DYNAMIC|PASSALL}]
DELETE FIREWALL POLICY=name INTERFACE=interface
```

An interface can only be defined as private in one security policy. A interface can only be defined as public in up to two security policies. Once at least one private interface and one public interface have been added, the firewall will be functional and will automatically implement the default policy rules:

- All flows originating from inside (i.e. private interfaces) are allowed. When a sessions is initiated from a private interface to an outside IP host and has been allowed by the firewall, traffic for that session can flow in both directions. When the session completes no further traffic is accepted to that private IP host on that port.
- All flows originating from the outside (i.e. public interfaces) are blocked.
- All traffic from an interface not specifically covered by policy, to an interface specified in a policy as private will be dropped.
- All traffic between interfaces not specifically covered by a policy will be passed as normal.

The current status and configuration of a policy or all policies can be displayed using the command:

```
SHOW FIREWALL POLICY=name [SUMMARY] [COUNTERS]
```

To further refine the control over flows to and from the public network, rules are added to the policy to allow or deny specific types of traffic.

Rules

Policy rules are used to refine the default security policy, which denies all access from hosts on the public network to hosts on the private network but allows all access from hosts on the private network to hosts on the public network.

Policy rules define precisely when and how traffic can flow through the firewall, based on IP addresses, port numbers, or protocol. For example, if a mail server is running on the private network, a rule could be added to allow TCP traffic to port 25 (the SMTP port) on the mail server host.

A rule is added to or deleted from a policy using the commands:

```
ADD FIREWALL POLICY=name RULE=rule-id ACTION={ALLOW|DENY}
INTERFACE=interface PROTOCOL={protocol|ALL|EGP|GRE|OSPF|
SA|TCP|UDP} [GBLIP=ipadd] [GBLPORT={ALL|port[-port]}]
[IP=ipadd[-ipadd]] [PORT={ALL|port[-port]|service-name}]
[REMOTEIP=ipadd[-ipadd]] [SOURCEPORT={ALL|port[-port]}]
DELETE FIREWALL POLICY=name RULE=rule-id
```

An existing rule can be modified using the command:

```
SET FIREWALL POLICY=name RULE=rule-id [PROTOCOL={protocol|
ALL|EGP|GRE|OSPF|SA|TCP|UDP}] [GBLIP=ipadd] [GBLPORT={ALL|
port[-port]}] [IP=ipadd[-ipadd]] [PORT={ALL|port[-port]|
service-name}] [REMOTEIP=ipadd[-ipadd]] [SOURCEPORT={ALL|
port[-port]}]
```

In addition to rules based on IP address, port and protocol, the processing of ICMP packets, IP packets with options set and ping packets can be enabled or disabled on a per-policy basis using the commands:

```
ENABLE FIREWALL POLICY=name [ICMP_FORWARDING={ALL|PARAMETER|
PING|REDIRECT|SOURCEQUENCH|TIMEEXCEEDED|TIMESTAMP|
UNREACHABLE}] [OPTIONS={ALL|RECORD_ROUTE|SECURITY|
SOURCEROUTE|TIMESTAMP}] [PING]

DISABLE FIREWALL POLICY=name [ICMP_FORWARDING={ALL|PARAMETER|
PING|REDIRECT|SOURCEQUENCH|TIMEEXCEEDED|TIMESTAMP|
UNREACHABLE}] [OPTIONS={ALL|RECORD_ROUTE|SECURITY|
SOURCEROUTE|TIMESTAMP}] [PING]
```

The currently configured rules for a policy can be displayed using the command:

```
SHOW FIREWALL POLICY=name
```

Rules are processed in order from the lowest number to the highest number. If rules both deny and allow an activity, the rule with the lowest number takes precedence. Typically, rules specify the access to or from a particular IP address and port.

Rules are processed as follows:

1. Based on the direction of the new flow or session, the default access result is set to the case of no matching rules. For new sessions or flows originating from a private network, access is set to *allowed*. For sessions and flows originating from a public network, access is set to *denied*. Each rule is then matched to the new flow or session until either a match is found or all rules have been rejected as not applicable, in which case the default access is used.
2. The protocol of the new flow is checked against the protocol field of the rule. If there is no match then the rule is rejected as not applicable.
3. The destination port is then matched to the rule port range. If there is no match then the rule is rejected as not applicable.
4. The source port is then matched to the rule's source port range if it is set. The source port used is dependent on the direction of the flow. For flows from a private network the source port of the flow is used. For flows from the public network, the destination port is used. If there is no match then the rule is rejected as not applicable.
5. The new flow's remote IP address is then matched to the rule's remote IP address or range if it is set. The remote IP address used is dependent on the direction of the flow. For flows from a private network the remote IP address used is the destination IP address of the flow. For flows from the public network, the source IP address if the flow is matched to the remote IP address of the rule. If there is no match then the rule is rejected as not applicable.
6. The new flow's IP address is matched to the rule's IP range or global IP address. If there is no match then the rule is rejected as not applicable. The IP address used is dependent on the direction of the flow. For flows from a private network the IP address used is the source IP address of the flow. For flows from the public network, the destination IP address is matched either to the IP address of the rule or to the global IP address set for the rule, depending on whether or not NAT is being applied to the interface.

If the rule action is ALLOW, then the new flow is allowed. If the rule action is to DENY then the flow is denied.

NAT

ENAT (*Enhanced NAT*) actually implements a form of dynamic packet filtering as a side effect of its implementation. To reduce the overhead of performing packet filtering twice (once by the firewall and once by NAT), the firewall has a built in NAT service that allows the IP addresses (and ports) of hosts on the private network to be translated using NAT or ENAT as they pass through the firewall.

A NAT translation is added to or removed from a policy using the commands:

```
ADD FIREWALL POLICY=name NAT={ENHANCED|STANDARD}
    INTERFACE=interface [IP=ipadd] GBLINTERFACE=interface
    [GBLIP=ipadd[-ipadd]]
DELETE FIREWALL POLICY=name NAT={ENHANCED|STANDARD}
    INTERFACE=interface GBLINTERFACE=interface [IP=ipadd]
```

Monitoring Firewall Activity

The firewall provides a range of options for monitoring the configuration of the firewall itself, as well as firewall events, access control and attacks.

Debugging

Debugging can be enabled or disabled on a per-policy basis using the commands:

```
ENABLE FIREWALL POLICY=name DEBUG={ALL|PACKET|PKT|PROCESS}
DISABLE FIREWALL POLICY=name DEBUG={ALL|PACKET|PKT|PROCESS}
```

Logging

The firewall can be configured to log an extensive range of events to the router's Logging Facility (Table 17-1 on page 17-6).

Table 17-1: Log types and subtypes for firewall events.

Option	Meaning
INATCP	Logs the start of TCP sessions initiated from the public Internet.
INAUDP	Logs the start of a UDP flow initiated from the public Internet.
INAICMP	Logs a ICMP request initiated from the public Internet.
INAOTHER	Logs the start of an IP protocol flow (other than TCP, UDP or ICMP) initiated from the public Internet.
INALLOW	Logs the start of all incoming allowed sessions and flows, and is the sum of the previous four values.
OUTATCP	Logs the start of TCP sessions initiated from the private Intranet.
OUTAUDP	Logs the start of a UDP flow initiated from the private Intranet.
OUTAICMP	Logs a ICMP request initiated from the private Intranet.
OUTAOTHER	Logs the start of an IP protocol flow (other than TCP, UDP or ICMP) initiated from the private Intranet.

Table 17-1: Log types and subtypes for firewall events. (Continued)

Option	Meaning
OUTALLOW	Logs the start of all allowed outgoing sessions and flows, and is the sum of the previous four values.
ALLOW	Logs the start of all allowed flows and sessions both in and out of the firewall.
INDTCP	Logs the failed start of TCP sessions initiated from the public Internet.
INDUDP	Logs the failed start of a UDP flow initiated from the public Internet.
INDICMP	Logs a failed ICMP request initiated from the public Internet.
INDOTHER	Logs the failed start of an IP protocol flow (other than TCP, UDP or ICMP) initiated from the public Internet.
INDENY	Logs the failed start of all denied incoming sessions and flows, and is the sum of the previous four values.
OUTDTCP	Logs the failed start of TCP sessions initiated from the private Intranet.
OUTDUDP	Logs the failed start of a UDP flow initiated from the private Intranet.
OUTDICMP	Logs a failed ICMP request initiated from the private Intranet.
OUTDOTHER	Logs the failed start of an IP protocol flow (other than TCP, UDP or ICMP) initiated from the private Intranet.
OUTDENY	Logs the failed start of all denied outgoing sessions and flows, and is the sum of the previous four values.
DENY	Logs the failed start of all flows and sessions both in and out of the firewall.
INDDTCP	Logs the failed start of TCP sessions initiated from the public Internet. Up to 192 bytes of the IP packet are also logged.
INDDUDP	Logs the failed start of a UDP flow initiated from the public Internet. Up to 192 bytes of the IP packet are also logged.
INDDICMP	Logs a failed ICMP request initiated from the public Internet. Up to 192 bytes of the IP packet are also logged.
INDDOTHER	Logs the failed start of an IP protocol flow (other than TCP, UDP or ICMP) initiated from the public Internet. Up to 192 bytes of the IP packet are also logged.
INDDUMP	Logs the failed start of all denied incoming sessions and flows, and is the sum of the previous four values. Up to 192 bytes of the IP packet are also logged.
OUTDDTCP	Logs the failed start of TCP sessions initiated from the private Intranet. Up to 192 bytes of the IP packet are also logged.
OUTDDUDP	Logs the failed start of a UDP flow initiated from the private Intranet. Up to 192 bytes of the IP packet are also logged.
OUTDDICMP	Logs a failed ICMP request initiated from the private Intranet. Up to 192 bytes of the IP packet are also logged.
OUTDDOTHER	Logs the failed start of an IP protocol flow (other than TCP, UDP and ICMP) initiated from the private Intranet. Up to 192 bytes of the IP packet are also logged.
OUTDDUMP	Logs the failed start of all denied OUT sessions and flows, and is the sum of the previous four values. Up to 192 bytes of the IP packet are also logged.
DENYDUMP	Logs the failed start of all flows and sessions both in and out of the firewall. Up to 192 bytes of the IP packet are also logged.

The logging of specific firewall events can be enabled or disabled on a per-policy basis using the commands:

```
ENABLE FIREWALL POLICY=name LOG={ALLOW|DENY|DENYDUMP|INAICMP|
  INALLOW|INAOOTHER|INATCP|INAUDP|INDDICMP|INDDOTHER|
  INDDTCP|INDDUDP|INDDUMP|INDENY|INDICMP|INDOTHER|INDTCP|
  INDUDP|OUTAICMP|OUTALLOW|OUTAOOTHER|OUTATCP|OUTAUDP|
  OUTDDICMP|OUTDDOTHER|OUTDDTCP|OUTDDUDP|OUTDDUMP|OUTDENY|
  OUTDICMP|OUTDOTHER|OUTDTCP|OUTDUDP}
DISABLE FIREWALL POLICY=name LOG={ALLOW|DENY|DENYDUMP|
  INAICMP|INALLOW|INAOOTHER|INATCP|INAUDP|INDDICMP|
  INDDOTHER|INDDTCP|INDDUDP|INDDUMP|INDENY|INDICMP|
  INDOTHER|INDTCP|INDUDP|OUTAICMP|OUTALLOW|OUTAOOTHER|
  OUTATCP|OUTAUDP|OUTDDICMP|OUTDDOTHER|OUTDDTCP|OUTDDUDP|
  OUTDDUMP|OUTDENY|OUTDICMP|OUTDOTHER|OUTDTCP|OUTDUDP}
```

Several options can be enabled or disabled in a single invocation by specifying the options as a comma separated list, for example:

```
ENABLE FIREWALL POLICY=office LOG=INDENY,OUTDENY
```

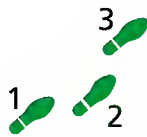
To minimise the number of log messages generated by the firewall, for some events the first four packets will be logged, then the first packet will be repeated with the text "(x *number*)" appended to indicate the number of repeat messages.

Configuration Examples

The following examples illustrate the steps required to configure the firewall for a range of applications. The configurations will provide very good firewall protection for a number of common router configurations. In particular, when a host on a network connected to a private interface initiates a session (TCP) or flow (UDP) to a host reachable by a public interface, then only context sensitive traffic relating to that session or flow is allowed back through the firewall. All other traffic initiated from hosts reachable by a public interface will be dropped by the firewall. The exception to this is when special filter rules have been added (see the fourth example below). Further, most common denial of service attacks will be logged and combated by the firewall.

Minimum Configuration for a Small Office

This example illustrates how to configure the most basic firewall for a small office wanting to be as secure as possible without restricting access to the public Internet. The office computers are connected to the router via Ethernet port 0, and there is a connection to the Internet via ISDN over PPP interface 0. The Ethernet interface has been assigned the global IP addresses 202.49.74.0 to 202.49.74.255. The PPP interface has been assigned a single global Internet address 202.49.72.2.



To configure a firewall without restricting access to the public Internet:

1. Create the security policy.

Create a policy named “office”, using the command:

```
CREATE FIREWALL POLICY=office
```

2. Add the interfaces to the security policy.

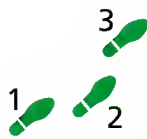
Add the Ethernet and PPP interfaces to the policy, using the commands:

```
ADD FIREWALL POLICY=office INTERFACE=eth0 TYPE=PRIVATE  
ADD FIREWALL POLICY=office INTERFACE=ppp0 TYPE=PUBLIC  
METHOD=DYNAMIC
```

Since externally initiated access to hosts on the private network is not required, no further configuration is required. When at least one private and one public interface are added to a policy, the policy is operational.

A Firewall with an ISP-assigned Internet Address

This example illustrates how to configure a firewall for a small office which is dynamically assigned a single global Internet address by their ISP when the router connects to the ISP and negotiates an IP option for the PPP link. For this reason NAT must be used on the private network. The office computers are connected to the router via Ethernet port 0, and there is a connection to the Internet via ISDN over PPP interface 0. The Ethernet interface will use the private IP network addresses 192.168.10.0 to 192.168.10.255. The PPP interface is dynamically assigned a single global Internet address by the ISP.



To configure Firewall with a single global Internet address from an ISP:

1. Create the security policy.

Create a policy named “office”, using the command:

```
CREATE FIREWALL POLICY=office
```

2. Add the interfaces to the security policy.

Add the Ethernet and PPP interfaces to the policy, using the commands:

```
ADD FIREWALL POLICY=office INTERFACE=eth0 TYPE=PRIVATE  
ADD FIREWALL POLICY=office INTERFACE=ppp0 TYPE=PUBLIC  
METHOD=DYNAMIC
```

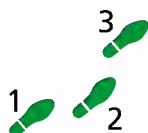
3. Add the NAT mapping to the private interface.

Add a NAT mapping to the Ethernet interface to translate private IP addresses to the dynamically assigned global IP address, using the command:

```
ADD FIREWALL POLICY=office NAT=ENHANCED INTERFACE=eth0  
GBLINTERFACE=ppp0
```

A Firewall with a Single Global Internet Address

This example is similar to the previous example, except that the ISP has assigned a single static global Internet address to the office. NAT must be used on the private network to translate private IP addresses to the global IP address. The office computers are connected to the router via Ethernet port 0, and there is a connection to the Internet via ISDN over PPP interface 0. The Ethernet interface will use the private IP network addresses 192.168.10.0 to 192.168.10.255. The PPP interface has been assigned the global Internet address 202.49.72.2.



To configure Firewall with a single global Internet address:

1. Create the security policy.

Create a policy named "office", using the command:

```
CREATE FIREWALL POLICY=office
```

2. Add the interfaces to the security policy.

Add the Ethernet and PPP interfaces to the policy, using the commands:

```
ADD FIREWALL POLICY=office INTERFACE=eth0 TYPE=PRIVATE
ADD FIREWALL POLICY=office INTERFACE=ppp0 TYPE=PUBLIC
METHOD=DYNAMIC
```

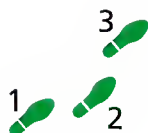
3. Add the NAT mapping to the private interface.

Add a NAT mapping to the Ethernet interface to translate private IP addresses to the statically assigned global IP address, using the command:

```
ADD FIREWALL POLICY=office NAT=ENHANCED INTERFACE=eth0
GBLINTERFACE=PPP0 GBLIP=202.49.72.2
```

Allowing Access to a WWW Server

This example builds on the previous example by allowing access from the public Internet to a WWW server on the private network. The office has been assigned a single global Internet address by their ISP. For this reason NAT must be used on the private network. The office computers are connected to the router via Ethernet port 0, and there is a connection to the Internet via ISDN over PPP interface 0. The Ethernet interface will use the private IP network addresses 192.168.10.0 to 192.168.10.255. The PPP interface has been assigned the single global Internet address 202.49.72.2. The office wants to provide access to a WWW server on the private network to advertise its products.



To configure Firewall to allow access to a WWW server:

1. Create the security policy.

Create a policy named "office", using the command:

```
CREATE FIREWALL POLICY=office
```

2. Add the interfaces to the security policy.

Add the Ethernet and PPP interfaces to the policy, using the commands:

```
ADD FIREWALL POLICY=office INTERFACE=eth0 TYPE=PRIVATE
ADD FIREWALL POLICY=office INTERFACE=ppp0 TYPE=PUBLIC
METHOD=DYNAMIC
```

3. Add the NAT mapping to the private interface.

Add a NAT mapping to the Ethernet interface to translate private IP addresses to the statically assigned global IP address, using the command:

```
ADD FIREWALL POLICY=office NAT=ENHANCED INTERFACE=eth0
GBLINTERFACE=PPP0 GBLIP=202.49.72.2
```

4. Add a rule to allow access to the WWW server.

The basic firewall configuration will not allow hosts on the private network to be accessed from the public network. To allow access to the office WWW server behind the firewall, add a rule to allow access to the WWW server at IP address 192.168.10.12 from the public Internet. Web browsers and web servers interact using the HTTP protocol, which is a TCP/IP-based protocol using a well-known port, so the rule must allow TCP traffic to the HTTP port to pass from the public interface to the private interface:

```
ADD FIREWALL POLICY=office RULE=1 ACTION=ALLOW
INTERFACE=ppp0 IP=192.168.10.12 PROTOCOL=TCP PORT=HTTP
GBLIP=202.49.72.2 GBLPORT=HTTP
```

Command Reference

This section describes the commands available on the router to enable, configure, control and monitor the firewall. The firewall requires IP to be enabled and configured correctly. See *Chapter 6, Internet Protocol (IP)* for the commands required to enable and configure IP.

See “Conventions” on page xxxv of *Preface* in the front of this manual for details of the conventions used to describe command syntax. See *Appendix A, Messages* for a complete list of messages and their meanings.

ADD FIREWALL POLICY INTERFACE

Syntax `ADD FIREWALL POLICY=name INTERFACE=interface TYPE={PUBLIC | PRIVATE} [METHOD={DYNAMIC | PASSALL}]`

where:

- *name* is a character string, 1 to 15 characters in length. Valid characters are letters (a–z, A–Z), digits (0–9) and the underscore character (“_”).
- *interface* is an interface name formed by concatenating a layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number in the range 0 to 3 (e.g. eth0, ppp1-1). If a logical interface is not specified, 0 is assumed (i.e. ‘eth0’ is equivalent to ‘eth0-0’).

Description This command adds an interface to the specified policy. The completed policy must contain at least one private interface and one public interface. An interface can only be specified as “private” in one policy. An interface can be specified as “public” in multiple policies. Multiple interfaces specified in a policy as “private” exchange packets without intervention from the firewall.

The **POLICY** parameter specifies the policy to which the interface will be added. The specified policy must already exist.

The **INTERFACE** parameter specifies an existing IP interface to be added to the policy.

The **TYPE** parameter specifies whether the interface is to be treated as a private interface (inside the firewall) or a public interface (outside the firewall).

The **METHOD** parameter specifies the method to be used by the firewall to pass packets between private and public interfaces, and is only valid if **TYPE** is set to **PUBLIC**. If **PASSALL** is specified, the firewall does not interfere with packet flow. This option should only be selected to allow an interface to run 1:1 NAT translation as defined in RFC 1631. If **DYNAMIC** is specified, dynamic packet filtering is used. The default is **DYNAMIC**.

Examples To add an interface to an existing policy named “zone1”, use the command:

```
ADD FIREWALL POLICY=zone1 INTERFACE=eth0 TYPE=PRIVATE
```

To add a WAN interface operating over PPP0 to the policy named “zone1”, use the command:

```
ADD FIREWALL POLICY=zone1 INTERFACE=PPP0 TYPE=PUBLIC
METHOD=PASSALL
```

See Also CREATE FIREWALL POLICY
DELETE FIREWALL POLICY INTERFACE
SHOW FIREWALL POLICY

ADD FIREWALL POLICY NAT

Syntax `ADD FIREWALL POLICY=name NAT={ENHANCED|STANDARD}
INTERFACE=interface [IP=ipadd] GBLINTERFACE=interface
[GBLIP=ipadd[-ipadd]]`

where:

- *name* is a character string, 1 to 15 characters in length. Valid characters are letters (a–z, A–Z), digits (0–9) and the underscore character (“_”).
- *ipadd* is an IP address in dotted decimal notation.
- *interface* is an interface name formed by concatenating a layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number in the range 0 to 3 (e.g. eth0, ppp1-1). If a logical interface is not specified, 0 is assumed (i.e. ‘eth0’ is equivalent to ‘eth0-0’).

Description This command adds a NAT translation to the specified policy. If an interface or global interface is specified then that interface must have already been added to the security policy.

The **POLICY** parameter specifies the policy to which the NAT translation will be added. The specified policy must already exist.

The **NAT** parameter specifies the type of NAT translation to perform. If **STANDARD** is specified, there is either a one-to-one translation between a private IP address and the specified global IP address, or if more than one

global IP address is supplied, then the global IP addresses are used dynamically from the supplied pool of addresses as required. When a pool of global addresses is specified and all sessions are complete for a particular global IP mapping, then that global IP address is returned to the pool for reuse. If ENHANCED is specified, Enhanced NAT (ENAT) is used and both the private IP address and protocol dependent port numbers are translated. The benefit of ENAT is that only a single global Internet address is required to map an entire private network.

The INTERFACE parameter specifies the private interface from which all received traffic is translated before being passed to the public interface specified by the GBLINTERFACE parameter. Both interfaces must already be defined and belong to the same policy.

The IP parameter specifies the private IP address used when a single public IP address is mapped to a single private IP address, and is only valid when NAT is set to STANDARD. This parameter is not valid if a range is specified for the GBLIP parameter.

The GBLINTERFACE parameter specifies the public interface from which all received traffic is translated before being passed to the private interface specified by the INTERFACE parameter. Both interfaces must already be defined and belong to the same policy.

The GBLIP parameter specifies a single global IP address or a range of global IP addresses to be used by the NAT translation. If NAT is set to STANDARD and a pool of global IP addresses is required then a range must be specified. If NAT is set to ENHANCED, then generally only a single global IP address is required. However, there are situations where it is necessary to allow sessions to be initiated from a public interface to private hosts via more than one public IP address. For example, WWW traffic for two public IP addresses that must be passed through to two private hosts. In this case, a range of global IP addresses is required. Only the first address of the range will be used as a source address for packets in outgoing sessions.

Examples To add an enhanced NAT mapping to the firewall policy named “zone1”, use the command:

```
ADD FIREWALL POLICY=zone1 NAT=ENHANCED INTERFACE=eth0
    GBLINTERFACE=PPP0 GBLIP=202.36.163.2
```

See Also CREATE FIREWALL POLICY
DELETE FIREWALL POLICY NAT
SHOW FIREWALL POLICY

ADD FIREWALL POLICY RULE

Syntax `ADD FIREWALL POLICY=name RULE=rule-id ACTION={ALLOW|DENY}
 INTERFACE=interface PROTOCOL={protocol|ALL|EGP|GRE|
 OSPF|SA|TCP|UDP} [GBLIP=ipadd] [GBLPORT={ALL|
 port[-port]}] [IP=ipadd[-ipadd]] [PORT={ALL|
 port[-port]|service-name} [REMOTEIP=ipadd[-ipadd]]
 [SOURCEPORT={ALL|port[-port]}]`

where:

- *name* is a character string, 1 to 15 characters in length. Valid characters are letters (a–z, A–Z), digits (0–9) and the underscore character (“_”).
- *rule-id* is a number in the range 1 to 299.
- *interface* is an interface name formed by concatenating a layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number in the range 0 to 3 (e.g. eth0, ppp1-1). If a logical interface is not specified, 0 is assumed (i.e. ‘eth0’ is equivalent to ‘eth0-0’).
- *protocol* is an Internet IP protocol number.
- *ipadd* is an IP addresses in dotted decimal notation.
- *port* is an Internet service port number or name.
- *service-name* is a predefined name for an IP service (Table 17-2 on page 17-15).

Description This command adds a rule defining the access allowed between private and public interfaces of the specified policy. By default all access from public interfaces (outside the firewall) is denied and all access from private interfaces (inside the firewall) is allowed. To refine the security policy additional rules can be added to allow or deny access based on IP addresses, port numbers, day of the week, or time of day. Each rule for a specific interface in a policy is processed in order, starting with the lowest numbered rule and proceeding to the highest numbered rule, or until a match is found.

In addition to rules based on IP address, port, protocol, date and time, the processing of ICMP packets, IP packets with options set and ping packets can be enabled or disabled on a per-policy basis using the ENABLE FIREWALL POLICY command on page 17-22 and the DISABLE FIREWALL POLICY command on page 17-21.

The POLICY parameter specifies the policy to which the rule will be added. The specified policy must already exist.

The RULE parameter specifies both an identifier for the rule and the position of the rule in the list of rules for this policy. Rules are processed in order, from the lowest to the highest numbered rule. The identifier is used to refer to this rule in other commands.

The ACTION parameter specifies whether the rule allows or denies a particular activity.

The INTERFACE parameter specifies the interface to which the rule will be applied. The interface must already exist and belong to the policy.

The PROTOCOL parameter specifies the IP protocol number or the name of a predefined protocol type to match. If TCP or UDP is specified, then the PORT parameter must also be specified.

The GBLIP parameter specifies a global IP address to be used as the public IP address for the rule if NAT is active on the interface.

The GBLPORT parameter specifies the port number, service name, or range of port numbers that apply to the rule if NAT is active on an interface.

The IP parameter specifies a single IP address or a range of IP addresses to match. If NAT is active on the interface, then the IP address range is that of the untranslated IP addresses.

The PORT parameter specifies a port number, a range of port numbers, or a predefined service name (Table 17-2 on page 17-15) to match. If ALL is specified, the rule matches any port number. If dynamic NAT is active on the interface it is possible to re-map a global port number to a different internal port number.

Table 17-2: Predefined IP protocol service names.

Service Name	Port Number
ECHO	7
DISCARD	9
FTP	21
TELNET	23
SMTP	25
TIME	37
DNS	53
BOOTPS	67
BOOTPC	68
TFTP	69
GOPHER	70
FINGER	79
WWW	80
HTTP	80
KERBEROS	88
RTELNET	107
POP2	109
POP3	110
SNMPTRAP	162
SNMP	161
BGP	179
RIP	520
VDOLIVE	7000
REALAUDIO	7070
REALVIDEO	7070

The REMOTEIP parameter specifies a single remote IP address or a range of remote IP addresses to match. This allows rules to be made based on the remote source of an IP flow.

The SOURCEPORT parameter specifies a source port to match for a TCP or UDP flow. This allows rules to be made based on the source port of the IP flow.

Examples To allow WWW access to an internal server at IP address 202.36.163.12, attached to a private interface defined in the policy named “zone1” via the public interface PPP0, use the command:

```
ADD FIREWALL POLICY=zone1 RULE=1 ACTION=ALLOW INTERFACE=ppp0
IP=202.36.163.12 PROTOCOL=TCP PORT=WWW
```

To allow DNS information from a server at 192.168.12.2 to a private DNS server at IP address 192.168.34.1, which uses UDP originating on port 53, use the command:

```
ADD FIREWALL POLICY=zone1 RULE=5 ACTION=ALLOW INTERFACE=ppp0
PROTOCOL=UDP IP=192.168.34.1 REMOTE=192.168.12.2
SOURCEPORT=53
```

To allow Telnet access to a UNIX server on a private network with NAT configured to use the public interface PPP0 with the global IP address 202.49.72.1, use the command:

```
ADD FIREWALL POLICY=zone1 RULE=6 ACTION=ALLOW INTERFACE=ppp0
IP=192.168.1.1 PROTOCOL=TCP PORT=TELNET GBLIP=202.49.72.1
GBLPORT=TELNET
```

See Also CREATE FIREWALL POLICY
DELETE FIREWALL POLICY RULE
SET FIREWALL POLICY RULE
SHOW FIREWALL POLICY

CREATE FIREWALL POLICY

Syntax CREATE FIREWALL POLICY=*name*

where:

- *name* is a character string, 1 to 15 characters in length. Valid characters are letters (a–z, A–Z), digits (0–9) and the underscore character (“_”).

Description This command creates a new firewall policy. The POLICY parameter specifies the name of the policy to be created, and is used in other commands to refer to the policy. The specified policy must not already exist.

A new policy will not become active until at least one private and one public interface have been added. The policy can be customised to handle specific traffic by adding interfaces, address lists, NAT translations and/or rules, using the commands:

```
ADD FIREWALL POLICY INTERFACE
ADD FIREWALL POLICY LIST
ADD FIREWALL POLICY NAT
ADD FIREWALL POLICY RULE
```

Examples To create a firewall policy named “area1”, use the command:

```
CREATE FIREWALL POLICY=area1
```

See Also ADD FIREWALL POLICY INTERFACE
ADD FIREWALL POLICY NAT
ADD FIREWALL POLICY RULE
DESTROY FIREWALL POLICY
DISABLE FIREWALL POLICY
ENABLE FIREWALL POLICY
SHOW FIREWALL POLICY

DELETE FIREWALL POLICY INTERFACE

Syntax DELETE FIREWALL POLICY=*name* INTERFACE=*interface*

where:

- *name* is a character string, 1 to 15 characters in length. Valid characters are letters (a–z, A–Z), digits (0–9) and the underscore character (“_”).
- *interface* is an interface name formed by concatenating a layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number in the range 0 to 3 (e.g. eth0, ppp1-1). If a logical interface is not specified, 0 is assumed (i.e. ‘eth0’ is equivalent to ‘eth0-0’).

Description This command deletes an interface from the specified policy. The resulting policy must contain at least one private interface and one public interface to remain operational.

The POLICY parameter specifies the policy from which the interface will be deleted. The specified policy must already exist.

The INTERFACE parameter specifies the interface to be deleted from the policy.

Examples To delete interface ETH0 from a policy named “zone1”, use the command:

```
DELETE FIREWALL POLICY=zone1 INTERFACE=eth0
```

See Also ADD FIREWALL POLICY INTERFACE
SHOW FIREWALL POLICY

DELETE FIREWALL POLICY NAT

Syntax `DELETE FIREWALL POLICY=name NAT={ENHANCED|STANDARD}
INTERFACE=interface GBLINTERFACE=interface [IP=ipadd]`

where:

- *name* is a character string, 1 to 15 characters in length. Valid characters are letters (a–z, A–Z), digits (0–9) and the underscore character (“_”).
- *interface* is an interface name formed by concatenating a layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number in the range 0 to 3 (e.g. eth0, ppp1-1). If a logical interface is not specified, 0 is assumed (i.e. ‘eth0’ is equivalent to ‘eth0-0’).
- *ipadd* is an IP address in dotted decimal notation.

Description This command deletes a NAT translation from an interface, or IP address associated with an interface.

The POLICY parameter specifies the policy from which the NAT translation or IP address will be deleted. The specified policy must already exist.

The NAT parameter specifies the type of NAT translation to be deleted. If STANDARD is specified, an IP address is not specified with the IP parameter, and a pool of global IP addresses exists, then the global IP address pool and the associated NAT translation are deleted. If STANDARD is specified and an IP address is specified with the IP parameter, the NAT translation for the specified private IP address is deleted. If ENHANCED is specified, the IP parameter may not be specified.

The INTERFACE parameter specifies the private interface for which the NAT translation will be deleted.

The GBLINTERFACE parameter specifies the public interface for which the NAT translation will be deleted.

The IP parameter specifies a previously defined private IP address used when a single public IP address is mapped to a single private IP address, for which the NAT translation will be deleted. The IP parameter is only valid when NAT is set to STANDARD.

Examples To delete a NAT mapping defined in the policy named “zone1”, use the command:

```
DELETE FIREWALL POLICY=zone1 NAT=ENHANCED INTERFACE=eth0  
GBLINTERFACE=ppp0
```

See Also ADD FIREWALL POLICY NAT
SHOW FIREWALL POLICY

DELETE FIREWALL POLICY RULE

Syntax `DELETE FIREWALL POLICY=name RULE=rule-id`

where:

- *name* is a character string, 1 to 15 characters in length. Valid characters are letters (a–z, A–Z), digits (0–9) and the underscore character (“_”).
- *rule-id* is a number in the range 1 to 299.

Description This command deletes a rule from the specified policy. The POLICY parameter specifies the policy from which the rule will be deleted. The specified policy must already exist. The RULE parameter specifies the rule to be deleted from the policy.

Examples To delete rule number 1 from the policy named “zone1”, use the command:

```
DELETE FIREWALL POLICY=zone1 RULE=1
```

See Also ADD FIREWALL POLICY RULE
SET FIREWALL POLICY RULE
SHOW FIREWALL POLICY

DELETE FIREWALL SESSION

Syntax `DELETE FIREWALL SESSION={session-number|ALL}`

where:

- *session-number* is the identifier for a currently active session.

Description This command terminates the specified currently active session or flow, or all currently active sessions and flows.

The SESSION parameter specifies the identifier of the active session or flow to be terminated. If ALL is specified, all active sessions and flows are terminated. The session identifier is read from the output of the SHOW FIREWALL SESSION command on page 17-32.

Examples To delete session number 1B32, use the command:

```
DELETE FIREWALL SESSION=1B32
```

See Also SHOW FIREWALL SESSION

DESTROY FIREWALL POLICY

Syntax DESTROY FIREWALL POLICY=*name*

where:

- *name* is a character string, 1 to 15 characters in length. Valid characters are letters (a–z, A–Z), digits (0–9) and the underscore character (“_”).

Description This command destroys the specified policy. The POLICY parameter specifies the policy to be destroyed. The specified policy must already exist.

Examples To destroy a policy named “area1”, use the command:

```
DESTROY FIREWALL POLICY=area1
```

See Also CREATE FIREWALL POLICY
DISABLE FIREWALL POLICY
ENABLE FIREWALL POLICY
SHOW FIREWALL POLICY

DISABLE FIREWALL

Syntax DISABLE FIREWALL

Description This command disables the firewall. A warning message, notification message and log message are generated when this command is issued.

Examples To disable the firewall, use the command:

```
DISABLE FIREWALL
```

See Also DISABLE FIREWALL POLICY
ENABLE FIREWALL
ENABLE FIREWALL POLICY
SHOW FIREWALL

DISABLE FIREWALL POLICY

Syntax `DISABLE FIREWALL POLICY=name [DEBUG={ALL|PACKET|PKT|PROCESS}] [ICMP_FORWARDING={ALL|PARAMETER|PING|SOURCEQUENCH|TIMEEXCEEDED|TIMESTAMP|UNREACHABLE}] [LOG={ALLOW|DENY|DENYDUMP|INAICMP|INALLOW|INAOTHER|INATCP|INAUDP|INDDICMP|INDDOTHER|INDDTCP|INDDUDP|INDDUMP|INDENY|INDICMP|INDOTHER|INDTCP|INDUDP|OUTAICMP|OUTALLOW|OUTAOTHER|OUTATCP|OUTAUDP|OUTDDICMP|OUTDDOTHER|OUTDDTCP|OUTDDUDP|OUTDDUMP|OUTDENY|OUTDICMP|OUTDOTHER|OUTDTCP|OUTDUDP}] [OPTIONS={ALL|RECORD_ROUTE|SECURITY|SOURCEROUTE|TIMESTAMP}] [PING]`

where:

- *name* is a character string, 1 to 15 characters in length. Valid characters are letters (a–z, A–Z), digits (0–9) and the underscore character (“_”).

Description This command disables the processing of specific types of IP packets by the specified policy, and/or disables logging or debugging for the policy.

The POLICY parameter specifies the policy for which packet processing attributes, accounting, logging or debugging are to be disabled. The specified policy must already exist.

The DEBUG parameter specifies the types of debugging information to be disabled. If ALL is specified, all debugging information is disabled. If PACKET or PKT is specified, the display of the first 56 bytes of each IP packet received is disabled. If PROCESS is specified, the display of information about the processing of a particular IP packet is disabled. The DEBUG parameter is not retained over a reboot.

The ICMP_FORWARDING parameter disables the forwarding of the specified ICMP messages through the router. The value may be a single option or a comma-separated list of options. The default is not to forward any ICMP messages because ICMP packets can be used as a method for denial of service attacks.

The LOG parameter disables the logging of the specified firewall events to the router’s Logging Facility. The value may be a single option or a comma-separated list of options. Table 17-1 on page 17-6 lists the options and their meanings.

The OPTIONS parameter disables the forwarding of packets with the specified IP option or options to the next level of firewall checking. The value may be a single option or a comma-separated list of options. The default is not to forward packets with IP options.

The PING parameter disables the handling of ping packets destined for the router itself. The default is to accept such ping packets.

Examples To disable the forwarding of all ICMP messages to the next level of firewall checking defined in the policy named “zone1”, use the command:

```
DISABLE FIREWALL POLICY=zone1 ICMP_FORWARDING=ALL
```

To disable the logging of all allowed sessions started from the public Internet, in the policy named "zone1", use the command:

```
DISABLE FIREWALL POLICY=zone1 LOG=INALLOW
```

See Also DISABLE FIREWALL
 DISABLE FIREWALL POLICY
 ENABLE FIREWALL
 ENABLE FIREWALL POLICY
 SHOW FIREWALL

ENABLE FIREWALL

Syntax ENABLE FIREWALL

Description This command enables the firewall. A log message is generated when this command is issued.

Examples To enable the firewall software, use the command:

```
ENABLE FIREWALL
```

See Also DISABLE FIREWALL
 DISABLE FIREWALL POLICY
 ENABLE FIREWALL POLICY
 SHOW FIREWALL

ENABLE FIREWALL POLICY

Syntax ENABLE FIREWALL POLICY=*name* [DEBUG={ALL|PACKET|PKT|PROCESS}] [ICMP_FORWARDING={ALL|PARAMETER|PING|SOURCEQUENCH|TIMEEXCEEDED|TIMESTAMP|UNREACHABLE}] [LOG={ALLOW|DENY|DENYDUMP|INAICMP|INALLOW|INAOOTHER|INATCP|INAUDP|INDDICMP|INDDOTHER|INDDTCP|INDDUDP|INDDUMP|INDENY|INDICMP|INDOTHER|INDTCP|INDUDP|OUTAICMP|OUTALLOW|OUTAOOTHER|OUTATCP|OUTAUDP|OUTDDICMP|OUTDDOTHER|OUTDDTCP|OUTDDUDP|OUTDDUMP|OUTDENY|OUTDICMP|OUTDOTHER|OUTDTCP|OUTDUDP}] [OPTIONS={ALL|RECORD_ROUTE|SECURITY|SOURCEROUTE|TIMESTAMP}] [PING]

where:

- *name* is a character string, 1 to 15 characters in length. Valid characters are letters (a–z, A–Z), digits (0–9) and the underscore character ("_").

Description This command enables the processing of specific types of IP packets by the specified policy, and/or enables logging or debugging for the policy.

The POLICY parameter specifies the policy for which packet processing attributes, accounting, logging or debugging are to be enabled. The specified policy must already exist.

The DEBUG parameter specifies the types of debugging information to be enabled. If ALL is specified, all debugging information is enabled. If PACKET or PKT is specified, the display of the first 56 bytes of each IP packet received is enabled. If PROCESS is specified, the display of information about the processing of a particular IP packet is enabled. The DEBUG parameter is not retained over a reboot.

The ICMP_FORWARDING parameter enables the forwarding of the specified ICMP messages through the router. The value may be a single option or a comma-separated list of options. The default is not to forward any ICMP messages because ICMP packets can be used as a method for denial of service attacks.

The LOG parameter enables the logging of the specified firewall events to the router's Logging Facility. The value may be a single option or a comma-separated list of options. Table 17-1 on page 17-6 lists the possible options and their meanings.

The OPTIONS parameter enables the forwarding of packets with the specified IP option or options to the next level of firewall checking. The value may be a single option or a comma-separated list of options. The default is not to forward packets with IP options.

The PING parameter enables the handling of ping packets destined for the router itself. The default is to accept such ping packets.

Examples To enable the passing of all ICMP messages to the next level of firewall checking defined in the policy named "zone1", use the command:

```
ENABLE FIREWALL POLICY=zone1 ICMP_FORWARDING=ALL
```

To enable the logging of all allowed sessions started from the public Internet and all denied sessions in both directions, in the policy named "zone1", use the command:

```
ENABLE FIREWALL POLICY=zone1 LOG=INALLOW,DENY
```

See Also DISABLE FIREWALL
DISABLE FIREWALL POLICY
ENABLE FIREWALL
SHOW FIREWALL

SET FIREWALL POLICY RULE

Syntax SET FIREWALL POLICY=*name* RULE=*rule-id* [PROTOCOL={*protocol* | ALL | EGP | GRE | OSPF | SA | TCP | UDP}] [GBLIP=*ipadd*] [GBLPORT={ALL | *port* [-*port*]}] [IP=*ipadd* [-*ipadd*]] [PORT={ALL | *port* [-*port*] | *service-name*}] [REMOTEIP=*ipadd* [-*ipadd*]] [SOURCEPORT={ALL | *port* [-*port*]}]

where:

- *name* is a character string, 1 to 15 characters in length. Valid characters are letters (a–z, A–Z), digits (0–9) and the underscore character (“_”).
- *rule-id* is a number in the range 1 to 299.
- *protocol* is an Internet IP protocol number.
- *ipadd* is an IP addresses in dotted decimal notation.
- *port* is an Internet service port number or name.
- *service-name* is a predefined name for an IP service (Table 17-2 on page 17-15).

Description This command modifies a rule defining the access allowed between private and public interfaces of the specified policy. By default all access from public interfaces (outside the firewall) is denied and all access from private interfaces (inside the firewall) is allowed. To refine the security policy additional rules can be added to allow or deny access based on IP addresses, port numbers, day of the week, or time of day. Each rule for a specific interface in a policy is processed in order, starting with the lowest numbered rule and proceeding to the highest numbered rule, or until a match is found.

In addition to rules based on IP address, port, protocol, date and time, the processing of ICMP packets, IP packets with options set and ping packets can be enabled or disabled on a per-policy basis using the ENABLE FIREWALL POLICY command on page 17-22 and the DISABLE FIREWALL POLICY command on page 17-21.

The POLICY parameter specifies the policy containing the rule to be modified. The specified policy must already exist.

The RULE parameter specifies the rule to be modified.

The PROTOCOL parameter specifies the IP protocol number or the name of a predefined protocol type to apply to the rule. If TCP or UDP is specified, then the PORT parameter must also be specified.

The GBLIP parameter specifies a global IP address to be used as the public IP address for the rule if NAT is active on the interface.

The GBLPORT parameter specifies the port number, service name, or range of port numbers that apply to the rule if NAT is active on an interface.

The IP parameter specifies a single IP address or a range of IP addresses to be applied by the rule. If NAT is active on the interface, then the IP address range is that of the untranslated IP addresses.

The PORT parameter specifies a port number, a range of port numbers, or a predefined service name (Table 17-2 on page 17-15) to match. If ALL is specified, the rule matches any port number. If dynamic NAT is active on the

interface it is possible to re-map a global port number to a different internal port number.

The REMOTEIP parameter specifies a single remote IP address or a range of remote IP addresses to be applied to the rule. This allows rules to be made based on the remote source of an IP flow.

The SOURCEPORT parameter specifies a source port for a TCP or UDP flow. This allows rules to be made based on the source port of the IP flow.

Examples To modify rule number 1 in the policy named “zone1” to match IP address 202.36.163.114, use the command:

```
SET FIREWALL POLICY=zone1 RULE=1 IP=202.36.163.114
```

See Also ADD FIREWALL POLICY RULE
DELETE FIREWALL POLICY RULE
SHOW FIREWALL POLICY

SHOW FIREWALL

Syntax SHOW FIREWALL

Description This command displays a summary of all security policies that have been created and the interfaces assigned to each policy (Figure 17-1 on page 17-25, Table 17-3 on page 17-25).

Figure 17-1: Example output from the SHOW FIREWALL command.

```
Firewall Configuration

Status ..... enabled

Policy : test
  Private Interface : eth0
  Public Interface : eth1
    Method ..... dynamic
    NAT ..... enhanced
      Method ..... enhanced dynamic
      Private Interface ..... eth0
      Global IP ..... 192.168.72.89
```

Table 17-3: Parameters displayed in the output of the SHOW FIREWALL command.

Parameter	Meaning
Status	The status of the firewall; one of “enabled” or “disabled”.
Policy	The name of a policy.
Private Interface	The name of a private interface assigned to the policy.
Public Interface	The name of a public interface assigned to the policy.
Method	The method used to packets to or from the public interface; one of “dynamic” or “passall”.

Table 17-3: Parameters displayed in the output of the SHOW FIREWALL command.

Parameter	Meaning
NAT	The type of NAT translation enabled; one of "standard" or "enhanced". Only displayed when NAT is enabled on the policy.
NAT/Method	The method used to perform NAT translation; one of "none", "static", "dynamic", "enhanced static", "enhanced dynamic" or "enhanced interface". This field depends on the combination of options configured in the ADD FIREWALL POLICY NAT command on page 17-12, and is only displayed when NAT is enabled on the policy.
NAT/Private Interface	The private interface to which NAT translations will apply. Only displayed when NAT is enabled on the policy.
NAT Global IP	The global IP address used by NAT translations. Only displayed when NAT is enabled on the policy.

See Also ADD FIREWALL POLICY INTERFACE
 CREATE FIREWALL POLICY
 DELETE FIREWALL POLICY INTERFACE
 DESTROY FIREWALL POLICY
 DISABLE FIREWALL
 ENABLE FIREWALL

SHOW FIREWALL POLICY

Syntax SHOW FIREWALL POLICY=*name* [COUNTERS] [SUMMARY]

where:

- *name* is a character string, 1 to 15 characters in length. Valid characters are letters (a–z, A–Z), digits (0–9) and the underscore character (" _").

Description This command displays detailed information about the specified or all policies (Figure 17-2 on page 17-27, Table 17-4 on page 17-27).

The POLICY parameter specifies the policy to be displayed. The specified policy must already exist. If a value is not specified then information for all policies is displayed.

The COUNTERS parameter displays counters for the specified policy or all policies (Figure 17-3 on page 17-29, Table 17-5 on page 17-30).

The SUMMARY parameter displays a summary of the information for each policy.

Figure 17-2: Example output from the SHOW FIREWALL POLICY command.

```

Policy : test
  Enabled Logging Options ..... allow denydump
  Enabled Debug Options ..... checksum
  Enabled IP options ..... none
  Enabled ICMP forwarding ..... ping timeexceeded
  Receive of ICMP PINGS ..... enabled
  Number of Active TCP Opens ..... 0
  Number of Active Sessions ..... 1
  Cache Hits ..... 429073
  Discarded ICMP Packets ..... 74
  Private Interface : eth0
    Rule ..... 1
      Action ..... allow
      Protocol ..... TCP
      Port ..... 23
      Global Port ..... all
  Public Interface : ppp0
    Method ..... dynamic
    NAT ..... enhanced
      Method ..... enhanced dynamic
      Private Interface ..... eth0
      Global IP ..... 192.168.72.89
    Rule ..... 2
      Action ..... allow
      IP ..... 202.36.163.20
      Protocol ..... TCP
      Port ..... 23
      Global IP ..... 192.168.72.89
      Global Port ..... 23

```

Table 17-4: Parameters displayed in the output of the SHOW FIREWALL POLICY command.

Parameter	Meaning
Policy	The name of a policy.
Enabled Logging Options	A list of the logging options currently enabled; one or more of "allow", "deny", "denydump", "inaicmp", "inallow", "inaother", "inatcp", "inaudp", "inddicmp", "inddother", "inddtcp", "inddudp", "indddump", "inddeny", "indicmp", "indother", "indtcp", "indudp", "outaicmp", "outallow", "outaother", "outatcp", "outaudp", "outddicmp", "outddother", "outddtcp", "outddudp", "outddump", "outdeny", "outdicmp", "outdoother", "outdtcp", "outdudp" or "none".
Enabled Debug Options	A list of the debug options currently enabled; one or more of "all", "packet", "process" or "none".
Enabled IP options	A list of the IP options allowed in IP packets to be forwarded by this policy; one or more of "all", "record_route", "security", "sourceroute", "timestamp" or "none".
Enabled ICMP forwarding	A list of the ICMP packet types that will be forwarded by this policy; one or more of "all", "parameter", "ping", "redirect", "sourcequench", "timeexceeded", "timestamp", "unreachable" or "none".
Receive of ICMP PINGS	Whether or not the reception of ICMP PING packets is enabled for this policy; one of "enabled" or "disabled".

Table 17-4: Parameters displayed in the output of the SHOW FIREWALL POLICY command. (Continued)

Parameter	Meaning
Number of Active TCP Opens	The number of currently active TCP connections for this policy.
Number of Active Sessions	The number of currently active sessions for this policy.
Cache Hits	The number of flow lookups found from the cache.
Discarded ICMP Packets	The number of ICMP packets discarded by this policy.
Private Interface	The name of a private interface assigned to the policy.
Public Interface	The name of a public interface assigned to the policy.
Method	The method used to packets to or from the public interface; one of "dynamic" or "passall".
NAT	The type of NAT translation enabled; one of "standard" or "enhanced". Only displayed when NAT is enabled on the policy.
NAT/Method	The method used to perform NAT translation; one of "none", "static", "dynamic", "enhanced static", "enhanced dynamic" or "enhanced interface". This field depends on the combination of options configured in the ADD FIREWALL POLICY NAT command on page 17-12, and is only displayed when NAT is enabled on the policy.
NAT/Private Interface	The private interface to which NAT translations will apply. Only displayed when NAT is enabled on the policy.
NAT Global IP	The global IP address used by NAT translations. Only displayed when NAT is enabled on the policy.
Rule	The identifier for a rule associated with the private or public interface.
Action	The action to perform when a flow matches this rule; one of "allow" or "deny".
Protocol	The IP protocol type to apply to this rule.
Port	The port number, service name (Table 17-2 on page 17-15) or range of port numbers to apply to this rule.
Global IP	The IP address to apply to this rule, if NAT is active on the interface.
Global Port	The port number, service name (Table 17-2 on page 17-15) or range of port numbers to apply to this rule, if NAT is active on the interface.
Remote IP	The remote IP address to match for this rule.
Source Port	The source port to match for this rule.

Figure 17-3: Example output from the SHOW FIREWALL POLICY COUNTERS command.

```

Policy : test
  Enabled Logging Options ..... allow denydump
  Enabled Debug Options ..... none
  Enabled IP options ..... none
  Enabled ICMP forwarding ..... ping timeexceeded
  Receive of ICMP PINGS ..... enabled
  Number of Active TCP Opens ..... 0
  Number of Active Sessions ..... 1
  Cache Hits ..... 430160
  Discarded ICMP Packets ..... 74
Private Interface : eth0
  Total Packets Received ..... 186331
  Number Flows Started ..... 9083
  Number Cache Hits ..... 173174
  Number Dropped Packets ..... 0
  Number Unknown IP Protocols ..... 0
  Number Bad ICMP Packets ..... 0
  Number Dumped ICMP Packets ..... 0
  Number Spoofing Packets ..... 0
  Number Dropped GBLIP is Zero .... 0
  Number No Spare Entries ..... 0
  Number FTP Port Commands ..... 0
  Number Bad FTP Port Commands .... 0
Public Interface : eth1
  Method ..... dynamic
  Total Packets Received ..... 264548
  Number Flows Started ..... 18
  Number Cache Hits ..... 256986
  Number Dropped Packets ..... 3751
  Number Unknown IP Protocols ..... 0
  Number Bad ICMP Packets ..... 0
  Number Dumped ICMP Packets ..... 0
  Number Spoofing Packets ..... 0
  Number Dropped GBLIP is Zero .... 0
  Number No Spare Entries ..... 0
  Number FTP Port Commands ..... 0
  Number Bad FTP Port Commands .... 0
NAT ..... enhanced
  Method ..... enhanced dynamic
  Private Interface ..... eth0
  Global IP ..... 192.168.72.89
Rule ..... 2
  Action ..... allow
  IP ..... 202.36.163.20
  Protocol ..... TCP
  Port ..... 23
  Global IP ..... 192.168.72.89
  Global Port ..... 23
  Number Hits ..... 0

```

Table 17-5: Parameters displayed in the output of the SHOW FIREWALL POLICY COUNTERS command.

Parameter	Meaning
Policy	The name of a policy.
Enabled Logging Options	A list of the logging options currently enabled; one or more of "allow", "deny", "denydump", "inaicmp", "inallow", "inaother", "inatcp", "inaudp", "inddicmp", "inddother", "inddtcp", "inddudp", "inddump", "indeny", "indicmp", "indother", "indtcp", "indudp", "outaicmp", "outallow", "outaother", "outatcp", "outaudp", "outddicmp", "outddother", "outddtcp", "outddudp", "outddump", "outdeny", "outdicmp", "outdother", "outdtcp", "outdudp" or "none".
Enabled Debug Options	A list of the debug options currently enabled; one or more of "all", "packet", "process" or "none".
Enabled IP options	A list of the IP options allowed in IP packets to be forwarded by this policy; one or more of "all", "record_route", "security", "sourceroute", "timestamp" or "none".
Enabled ICMP forwarding	A list of the ICMP packet types that will be forwarded by this policy; one or more of "all", "parameter", "ping", "redirect", "sourcequench", "timeexceeded", "timestamp", "unreachable" or "none".
Receive of ICMP PINGS	Whether or not the reception of ICMP PING packets is enabled for this policy; one of "enabled" or "disabled".
Number of Active TCP Opens	The number of currently active TCP connections for this policy.
Number of Active Sessions	The number of currently active sessions for this policy.
Cache Hits	The number of flow lookups found from the cache.
Discarded ICMP Packets	The number of ICMP packets discarded by this policy.
Private Interface	The name of a private interface assigned to the policy.
Public Interface	The name of a public interface assigned to the policy.
Total Packets Received	The total number of packets received on the interface.
Number Flows Started	The number of flows started on the interface.
Number Cache Hits	The number of flow lookups for the interface found from the cache.
Number Dropped Packets	The number of packets received on the interface that were dropped.
Number Unknown IP Protocols	The number of packets received on the interface with an unknown IP protocol.
Number Bad ICMP Packets	The number of badly formatted ICMP packets received on the interface.
Number Dumped ICMP Packets	The number of ICMP packets received on the interface that were dumped.
Number Spoofing Packets	The number of Smurf attack packets received on the interface.
Number Dropped GBLIP Zero	The number of packets received on the interface that were dumped because the global IP address was zero.
Number No Spare Entries	The number of packets received on the interface that were dumped because the system had insufficient memory.

Table 17-5: Parameters displayed in the output of the SHOW FIREWALL POLICY COUNTERS command. (Continued)

Parameter	Meaning
Number FTP Port Commands	The number of valid FTP port commands received on the interface.
Number Bad FTP Port Commands	The number of invalid FTP port commands received on the interface.
Method	The method used to packets to or from the public interface; one of "dynamic" or "passall".
NAT	The type of NAT translation enabled; one of "standard" or "enhanced". Only displayed when NAT is enabled on the policy.
NAT/Method	The method used to perform NAT translation; one of "none", "static", "dynamic", "enhanced static", "enhanced dynamic" or "enhanced interface". This field depends on the combination of options configured in the ADD FIREWALL POLICY NAT command on page 17-12, and is only displayed when NAT is enabled on the policy.
NAT/Private Interface	The private interface to which NAT translations will apply. Only displayed when NAT is enabled on the policy.
NAT Global IP	The global IP address used by NAT translations. Only displayed when NAT is enabled on the policy.
Rule	The identifier for a rule associated with the private or public interface.
Action	The action to perform when a flow matches this rule; one of "allow" or "deny".
Protocol	The IP protocol type to apply to this rule.
Port	The port number, service name (Table 17-2 on page 17-15) or range of port numbers to apply to this rule.
Global IP	The IP address to apply to this rule, if NAT is active on the interface.
Global Port	The port number, service name (Table 17-2 on page 17-15) or range of port numbers to apply to this rule, if NAT is active on the interface.
Remote IP	The remote IP address to match for this rule.
Source Port	The source port to match for this rule.

See Also

- ADD FIREWALL POLICY INTERFACE
- ADD FIREWALL POLICY NAT
- ADD FIREWALL POLICY RULE
- CREATE FIREWALL POLICY
- DELETE FIREWALL POLICY INTERFACE
- DELETE FIREWALL POLICY NAT
- DELETE FIREWALL POLICY RULE
- DESTROY FIREWALL POLICY
- DISABLE FIREWALL POLICY
- ENABLE FIREWALL POLICY
- SET FIREWALL POLICY RULE
- SHOW FIREWALL

SHOW FIREWALL SESSION

Syntax `SHOW FIREWALL SESSION[=session-number] [POLICY=name]
[COUNTERS] [PORT={port-port|service-name}]
[PROTOCOL={protocol|ALL|EGP|ICMP|OSPF|TCP|UDP}]
[SUMMARY]`

where:

- *session-number* is the identifier for a currently active session.
- *name* is a character string, 1 to 15 characters in length. Valid characters are letters (a–z, A–Z), digits (0–9) and the underscore character (“_”).
- *port* is an Internet service port number or name.
- *service-name* is a predefined name for an IP service (Table 17-2 on page 17-15).
- *protocol* is an Internet IP protocol number.

Description This command displays information about the sessions and flows currently active for the specified policy (Figure 17-4 on page 17-32). If SESSION is specified, only information about the specified session is displayed. Otherwise, information about all sessions is displayed.

The POLICY parameter specifies the policy for which session information is to be displayed. The specified policy must already exist. If a value is not specified, session information for all policies is displayed.

If COUNTERS is specified, session counters for the specified policy are displayed.

If SUMMARY is specified, only summary information for the specified policy is displayed.

If PROTOCOL is specified, the display is limited to sessions based on the specified IP protocol type.

If PORT is specified, the display is limited to sessions between ports in the specified range of ports or using the specified service (Table 17-2 on page 17-15).

Figure 17-4: Example output from the SHOW FIREWALL SESSION command.

```
Policy : test
Current Sessions
-----
cc2b TCP 202.36.163.10:1383 192.168.72.89:52267 192.168.72.50:21
TCP state ..... established
Start time ..... 17:58:57 19-Apr-1999
Minutes to deletion ..... 536
-----
```

Table 17-6: Parameters displayed in the output of the SHOW FIREWALL SESSION command.

Parameter	Meaning
Policy	The name of a policy.
<i>hex-num</i>	The session identifier
TCP/UDP/ <i>number</i>	The IP protocol (one of "TCP", "UDP" or an IP protocol number), followed by the source address:port, the global IP address:mapped port, and the destination IP address:port
Packets from private IP	The number of packets forwarded from the private network to the public network.
Octets from private IP	The number of octets forwarded from the private network to the public network.
Packets to private IP	The number of packets forwarded from the public network to the private network.
Octets to private IP	The number of octets forwarded from the public network to the private network.
TCP state	The state of the TCP session; one of "free", "closed", "listen", "synSent", "synReceived", "established", "finWait1", "finWait2", "closeWait", "lastAck", "closing", "timeWait", "deleteTCB", "synSent" or "synReceived" .
Private SEQ number	The current sequence number for the TCP connection to the private IP address.
Private ACK number	The current acknowledgement number for the TCP connection to the private IP address.
Private max window size	The current maximum window size for the TCP connection to the private IP address.
Public SEQ number	The current sequence number for the TCP connection to the public IP address.
Public ACK number	The current acknowledgement number for the TCP connection to the public IP address.
Public max window size	The current maximum window size for the TCP connection to the public IP address.
Sequence Delta	The different between the current sequence numbers for the private and public connections.
ICMP type	The type of ICMP request, for ICMP sessions; one of "Echo request", "Time request", "Name request" or "Unknown ICMP type" .
Start time	The date and time that the session was started.
Minutes to deletion	The number of minutes remaining before the session is automatically deleted.

See Also DELETE FIREWALL SESSION
SHOW FIREWALL POLICY

Chapter 18

Link Compression

Introduction	18-2
Overview	18-2
Link Compression	18-2
PPP	18-3
X.25	18-4

Introduction

This chapter describes the link compression facilities on the router, and how to configure the Point-to-Point Protocol (PPP) and X.25 link layers to use link compression.

Overview

Link compression has traditionally been provided by external devices connected between a router port and the WAN access device. The disadvantages of external compression devices is that they require a separate connection to each router port requiring compression and to each WAN access device, they can not be managed from within the router's management structure, and they normally don't support dial-up interfaces such as ISDN.

Integrating compression functions into the router enables a single compression resource to support the compression of multiple links over any router interface, replacing multiple external devices. Integration also allows the router to support protocols such as PPP multilink, which can spread data from one compression channel across multiple physical links. The compression process can be configured and monitored using the router's own management interface, instead of a separate management system used only for the external device.

Link compression operates by compressing the whole data stream, including the network layer packet headers used for routing. This means that the packet header is no longer accessible by intermediate routers which do not support the compression algorithm used. Even if an intermediate router does support the compression algorithm, packets must be decompressed, then compressed again at each router so that the packet headers can be read. Consequently, external link compression is normally only used in point-to-point configurations where the local and remote routers are directly connected, without any intermediate routers.

Link Compression

The ENCO module provides multichannel link compression allowing multiple higher layer modules to use compression simultaneously. STAC LZS compression is available in software. To configure the ENCO module for software compression See *Chapter 8, Compression Services*.

Link compression is supported on PPP and X.25 interfaces. Routing modules such as IP can gain access to the data compression facilities via these link level protocols. PPP can be configured on a per-interface or per-link basis to use compression. Configuring PPP compression on a per-link basis allows some links in a multilink bundle to be compressed while other links are uncompressed. This has advantages in situations where compression is already being provided on some links, for example on links provided by compressing modems. X.25 is configured on a per-circuit basis.

When a packet is passed to the compression facility for compression, the entire packet following the PPP or X.25 header is compressed, resulting in a high compression efficiency since the header of any higher layer protocol being

carried by these protocols is compressed along with the data. This means that if TCP/IP is being transported over a compressed PPP link then the TCP/ IP header will be compressed.

PPP

The router implements the Compression Control Protocol (CCP) as defined by RFC 1962 to provide link compression on PPP interfaces. CCP provides a method for negotiating the compression algorithm to use and algorithm-specific parameters such as the check mode. It also provides a mechanism for synchronising the compression histories at each end of the link if they become unsynchronised. The use of STAC LZS compression with CCP is defined in RFC 1962.

PPP commands are used to enable compression for each PPP interface. For example, to create PPP interface 0 over ISDN call “Office” and enable STAC LZS compression on the link, use the command:

```
CREATE PPP=0 OVER=ISDN-Office COMPRESSION=ON  
COMPALGORITHM=STAC STACCHECK=LCB
```

PPP negotiates with the router at the remote end of the link; if both routers have compression enabled on the link and can agree on a compression algorithm, compression will be used on the link.

During CCP negotiation the router offers the remote router a choice of compression algorithms that it is prepared to decompress. The router only offers a compression algorithm if there are channels configured and available for that algorithm. Only the STAC LZS compression algorithm (CCP option 17) is supported. The peer then chooses the option it prefers to compress packets with and informs the offering router of its choice so that it can configure its decompressing channel. If all compression options are rejected by the peer then packets are sent uncompressed.

As part of the negotiation of the STAC LZS option, the router negotiates to add an 8-bit Longitudinal Check Byte (LCB) or an 8-bit sequence number to the data, enabling the packet to be validated during decompression. A check value is useful because PPP does not guarantee reliable, in-order delivery of packets. If a packet is corrupted or lost then when the next packet is received it's decompression will fail because the compression histories will be out of step. Adding a check value allows unsuccessful decompressions to be detected.



To use Software Release 7.4 STAC LZS compression with Software Release 7.2 STAC LZS compression, the check mode on the router running Software Release 7.4 must be set to LCB. To use a COMP-2 coprocessor engine to provide STAC LZS compression, the check mode must be set to LCB at both ends of the PPP link.

When compression histories become unsynchronised while using STAC LZS compression a *Reset Request—Reset Ack* protocol, described in RFC 1962, is used to resynchronise the compression histories.

For more information about configuring PPP link compression see *Chapter 3, Point-to-Point Protocol (PPP)*.

X.25

The router uses a simple static configuration process to provide STAC LZS link compression for X.25. X.25 does not reset compression links as it is a reliable transmission protocol. Link compression on X.25 interfaces is configured on a per-circuit basis. For example, to add a MIOX circuit named "RemoteOffice" to X.25T interface 0 and enable compression, use the command:

```
ADD MIOX=1 CIRCUIT=RemoteOffice PVC=1 COMP=ON
```

For more information about configuring X.25 compression see *Chapter 5, X.25*.

[K](#)

Appendix A

Messages

Introduction	A-2
Message Descriptions	A-2
smm001–smm255: Global Messages	A-2
s03256–s03999: Point-to-Point Protocol	A-7
s05256–s05999: Internet Protocol (IP)	A-10
s14256–s14999: Q.931	A-15
s18256–s18999: TEST Module	A-16
s19256–s19999: LAPD	A-18
s22256–s22999: TCP	A-20
s23256–s23999: Ethernet Driver	A-20
s28256–s28999: Compression	A-21
s30256–s30999: X.25 Layer 3 (DTE)	A-22
s31256–s31999: FLASH Driver	A-24
s33256–s33999: TELNET	A-25
s34256–s34999: System	A-25
s35256–s35999: Command Processor	A-26
s36256–s36999: TTY	A-26
s37256–s37999: ISDN Call Control	A-27
s38256–s38999: MIOX	A-30
s39256–s39999: BOOTP	A-31
s41256–s41999: BRI Driver	A-32
s43256–s43999: PORT Driver	A-33
s45256–s45999: User Authentication Facility	A-35
s48256–s48999: LOADER	A-38
s49256–s49999: INSTALL	A-40
s53256–s53999: Trigger Facility	A-41
s54256–s54999: Scripting	A-43
s55256–s55999: Time Division Multiplexing (TDM)	A-43
s56256–s56999: File Subsystem	A-45
s57256–s57999: Logging Facility	A-46
s58256–s58999: PING	A-49
s59256–s59999: Simple Network Management Protocol (SNMP)	A-50
s61256–s61999: Telephony Services	A-51
s70256–s70999: Dynamic Host Configuration Protocol (DHCP)	A-52
s77256–s77999: Firewall	A-53

Introduction

This appendix contains a complete list, in numeric order, of all router messages, and their descriptions.

The general format of router messages is:

Severity (message-number) : Message-text.

where:

- *Severity* is the severity level of the message, and will be one of the words:
Info — the operation was successful and the message contains information.
Warning — the operation was successful, but some warning is required.
Error — the operation was not successful.
- *message-number* is the message number, which uniquely identifies the message.
- *Message-text* is the text of the message, which may include variable parts.

The message number is formatted as:

smmmnnn

where:

- *s* is a single decimal digit representing the severity level of the message and will be one of 0 (Info), 1 (Warning) or 2 (Error).
- *mm* is a two-digit decimal number representing the module which generated the message.
- *nnn* is a three-digit decimal number representing the module-specific message number.

Message Descriptions

smm001–smm255: Global Messages

smm000: Parameter <string> seen twice

While parsing a command, the specified parameter was seen twice, but it may only be specified once. Re-enter the command, specifying the parameter only once.

smm001: Internal error

While processing a command, an internal error was detected. This should be reported to the router manufacturer.

smm002: Specified interface is not defined

The interface specified in the command is not currently defined. Either define the interface first and then re-enter the command, or re-enter the command specifying another interface that is defined.

smm003: Operation successful

The command just entered was executed successfully.

smm004: Invalid interface

The interface specified in the command is invalid.

smm005: No value allowed on parameter <string>

A value was given for the specified parameter, but the parameter does not accept values. Re-enter the command and without a value for the parameter.

smm006: Syntax error, <string>

A syntax error was detected while parsing a parameter. The type of error is displayed. Re-enter the command with the correct syntax.

smm007: Unexpected end of line

The end of the command line was reached before the command processor expected it. A parameter or parameters may be missing. Re-enter the command with the correct parameters.

smm008: Extra parameters on command line

The command handler found more parameters on the command line than expected. Re-enter the command with the correct number and combination of parameters.

smm009: Internal error: <number>

The command handler detected an internal error. Report this message to your distributor or reseller, along with the number in the message, the release of software being run and the command line as entered.

smm010: Value missing on parameter <string>

The parameter specified requires a value. Either the parameter has been entered incorrectly or the value required has not been entered. Re-enter the command correctly.

smm011: Value not allowed on parameter <string>

The parameter specified may not have a value associated with it. Either the parameter has been entered incorrectly or a value has been given incorrectly. Re-enter the command correctly.

smm012: Parameter "<string>" not recognised

The string in quotes was not recognised as a parameter for this command. Either an invalid parameter was entered or the parameter was spelt incorrectly. Re-enter the command with the correct parameter or the correct spelling.

smm013: Privilege violation on parameter <string>

The parameter specified requires a higher privilege than that currently assigned to the terminal from which the command was entered. Either login to the router with a user name that has a higher privilege (e.g. MANAGER) or re-enter the command without the parameter.

smm014: Privilege violation on parameter "<string>"

The parameter specified requires a higher privilege than that currently assigned to the terminal from which the command was entered. Either login to the router with a user name that has a higher privilege (e.g. MANAGER) or re-enter the command without the parameter.

smm015: Parameter <string>, invalid decimal integer "<string>"

The parameter specified accepts a value expressed as a decimal integer but the value entered was not recognised as such. Re-enter the command with a valid decimal integer as the value of this parameter.

smm016: Parameter <string>, value too low; minimum is <number>

The value entered for the specified parameter is too low. The minimum allowable value is displayed. Re-enter the command with a new value within the allowed range.

smm017: Parameter <string>, value too high; maximum is <number>

The value entered for the specified parameter is too high. The maximum allowable value is displayed. Re-enter the command with a new value within the allowed range.

smm018: Parameter <string>, invalid HEX integer "<string>"

The parameter specified accepts a value expressed as a HEX integer but the value entered was not recognised as such. Re-enter the command with a valid HEX integer as the value of this parameter.

smm019: Parameter <string>, value too low; minimum is <hexnum>

The value entered for the specified parameter is too low. The minimum allowable value is displayed. Re-enter the command with a new value within the allowed range.

smm020: Parameter <string>, value too high; maximum is <hexnum>

The value entered for the specified parameter is too high. The maximum allowable value is displayed. Re-enter the command with a new value within the allowed range.

smm021: Parameter <string>, string too short; minimum length is <number>

The string entered for the specified parameter is too short. The minimum length is displayed. Re-enter the command with a string within the allowable length limits.

smm022: Parameter <string>, string too long; maximum length is <number>

The string entered for the specified parameter is too long. The maximum length is displayed. Re-enter the command with a string within the allowable length limits.

smm023: Parameter <string>, string contains invalid character "<char>"

The specified parameter was entered with a string value which contains an invalid character. Re-enter the command with only valid characters in the string.

smm024: Parameter <string>, invalid Ethernet address "<string>"

The specified parameter accepts a value that is an Ethernet address but the value entered was ill-formed or illegal. Re-enter the command with a valid Ethernet address for this parameter.

smm025: Parameter <string>, invalid IP address or netmask "<string>"

The specified parameter accepts a value that is an IP address or an IP network mask but the value entered was ill-formed or illegal. Re-enter the command with a valid IP address or network mask for this parameter.

smm027: Parameter <string>, invalid X.121 address "<string>"

The specified parameter accepts a value that is an X.121 address but the value entered was ill-formed or illegal. Re-enter the command with a valid X.121 address for this parameter.

smm029: Parameter <string>, not a single character

The specified parameter accepts a value that is a single character, but more than one character was entered. Re-enter the command with a single character for this parameter.

smm030: Parameter <string>, invalid character "<char>"

The specified parameter accepts a value that is a single character, but the character entered is invalid. Re-enter the command with a valid character for this parameter.

smm031: Parameter <string>, value "<string>" unrecognised

The value entered for the specified parameter was unrecognised. Re-enter the command with a value that is recognised for this parameter.

smm032: Parameter <string>, privilege violation on value <string>

The value entered for the specified parameter was recognised, but the privilege level of the terminal from which the command was entered was not high enough for the value. Re-enter the command with a value that is of the correct privilege or set the privilege level of the terminal high enough for the value.

smm033: Parameter <string>, invalid value or decimal integer "<string>"

The specified parameter accepts a value that is either a well-known value or a decimal integer, but neither was recognised.

smm039: Parameter <string>, invalid value or HEX integer "<string>"

The specified parameter accepts a value that is either a well-known value or a HEX integer, but neither was recognised.

smm040: Parameter <string>, invalid value, IP address or netmask "<string>"

The specified parameter accepts a value that is either a well-known value, an IP address or an IP network mask, but none of these options was recognised.

smm041: Parameter <string>, <string>, module doesn't support interface

The interface specified is of a type not supported by the module you are trying to use it with.

smm043: Parameter <string>, two range separators seen

In entering a range of values for the specified parameter, two range separators (the character that comes between the two values) were seen. Re-enter the command with only one range separator.

smm044: Parameter <string>, <string> end of range not defined

A range of values was entered for the specified parameter, but the lower or upper (given) end of the range was specified by not supplying a value. This is not valid for the given end of the range. Re-enter the command, but specify the end of the range.

smm045: <module> - module is now disabled

The given module has now been disabled. This message is produced as the result of a PURGE <module> command.

smm046: Parameters <string> mutually exclusive

The parameters in the error message are mutually exclusive, that is, they can not be entered in the same command. Re-enter the command leaving one or more parameters off.

smm047: Parameter <string> requires parameter(s) <string>

The first parameter in the message was present in the command, but requires the presence of one or more other parameters, also given in the message. Re-enter the command, specifying all required parameters.

smm048: Parameter <string>, invalid separator "<string>"

The given parameter had an invalid separator in it. The invalid separator is given. Possible separators are =, !=, >, >=, < and <=, but not all of these are valid for every parameter. Re-enter the command giving a valid separator for the parameter.

smm049: The <module> module is not enabled

The specified module is not enabled, so the operation can not be carried out. Some modules in the router require that the module be enabled before they can be configured. Re-enter the command after enabling the module.

smm050: The <module> module has been reset

The specified module has been reset due to a RESET <module> command.

smm051: The <module> module has been purged

The specified module has been purged due to a PURGE <module> command. All configuration for the module has been lost.

smm052: The <module> module has been enabled

The specified module has been enabled. Operations with this module can now proceed. In many cases the module must be enabled in order to configure it.

smm053: The <module> module has been disabled

The specified module has been disabled. Operations with this module have ceased.

smm054: The <module> module is already enabled

An attempt to enable the module was made, but the module is already enabled. The command was redundant.

smm055: The <module> module is already disabled

An attempt to disable the module was made, but the module is already disabled. The command was redundant.

smm056: <string> has been disabled

A particular specified feature in the router has been disabled. This is an informational message.

smm057: <string> has been enabled

A particular specified feature in the router has been enabled. This is an informational message.

smm058: No <string> currently available or configured

There are none of the specified objects available or configured, so nothing to display. This is an informational/warning message.

smm059: Parameter <string> required but not seen in command line

The given parameter was required in the command, but was not seen. Re-enter the command with the required parameter included.

smm060: Nothing to display

The SHOW command just entered resulted in no output. What was being showed does not exist, or has not been configured.

smm061: <string> not found

The specified object was not found. Re-enter the command specifying an object that actually exists.

smm062: <string> <string> not found

The object of a given name or number was not found. Re-enter the command specifying an object that actually exists.

smm063: <string> already exists

The specified object already exists. Re-enter the command specifying an object that doesn't already exist.

smm064: Router is in secure mode. Command needs Security Officer privilege.

The command specified requires a Security Officer privilege to be assigned to the terminal from which the command was entered. Login to the router with a user name that has Security Officer privilege.

smm065: Security Officer privilege timed out on command <string>. Please re-login.

The command specified requires a Security Officer privilege and it has been more than ten minutes since a Security Officer privilege command was entered, so the user needs to login to regain Security Officer privilege.

smm066: Parameter <string> has bad syntax

The parameter specified has been entered with incorrect syntax. The command must be re-entered with the parameter entered with correct syntax. This message is reserved for parameters which have complex syntax to cover any syntax problems found in parsing the parameter.

s03256–s03999: Point-to-Point Protocol**s03256: Parameter <string>, invalid interface <string>**

The specified parameter should represent a valid interface on which PPP can run but doesn't. Re-enter the command with a valid interface for the parameter.

s03257: Specified PPP interface already exists

The specified PPP interface already exists, so it can not be created again. Re-enter the command and specify an interface that does not already exist.

s03258: Parameter <string>, too long

The username or password parameters have a maximum length of 15 characters. Re-enter the command with a shorter username or password.

s03259: Specified PPP interface does not exist

The specified PPP interface does not exist, so it can not be altered or destroyed. Re-enter the command with an interface that does exist.

s03260: OVER parameter required with this command

The OVER parameter is required with this command. Re-enter the command with the OVER parameter, specifying a lower layer interface.

s03261: <string> parameter not allowed with this lower layer interface type

The parameter is only valid when the lower layer interface is an ISDN call.

s03263: Lower layer interface not found

The lower layer interface specified in the OVER parameter can not be found. Check that the interface exists.

s03264: OVER parameter required with these parameters

The OVER parameter is required with at least one of the specified parameters in the command. Refer to the Reference Manual for the parameters to which this applies.

s03265: Only one parameter allowed.

Only one parameter can be specified at a time. Re-enter the command with only one parameter.

s03266: No PPP interfaces found

For SHOW commands, there were no PPP interfaces so no information can be displayed.

s03267: Can not perform this operation on a dynamic interface

The PPP interface was created dynamically and therefore the ADD and DELETE operations can not be performed on it.

s03268: <string> <string> is already in use

An attempt was made to configure a PPP interface over an interface that is already being used.

s03269: <string> <string> does not exist

An attempt was made to configure a PPP interface over an interface that does not exist.

s03270: The NUMBER parameter is only valid for ISDN calls

The NUMBER parameter was used in a DELETE command for an interface other than an ISDN call. The NUMBER parameter only applies to ISDN calls.

s03271: The link is not a <string> link

The specified link is not of the type specified in the command.

s03272: <string> parameter requires COMPRESSION set to <string>

The COMPALGORITHM or STACCHECK parameters have been specified without the COMPRESSION parameter set to ON or LINK.

s03273: Can only set <string> when <string> is set to <string>

An attempt was made to set the COMPALGORITHM or STACCHECK parameters when neither interface compression nor link compression were on.

s03274: Parameter COMPRESSION, value "on" unrecognised

The COMPRESSION parameter was set to ON in an ADD command.

s03276: TDM group <string> has no slots configured

An attempt was made to create a PPP interface over a TDM group that has no slots configured.

s03277: Multilink is not active

An attempt was made to show the multilink counters when multilink is not active.

s03278: Interface compression is on

An attempt was made to set link compression when interface compression is on. Need to set interface compression off before setting link compression.

s03281: Callback <string> to set <string>

An attempt was made to set a callback parameter while in the wrong callback mode.

s03282: Authentication must be enabled when accepting callback

An attempt was made to set the callback mode to ACCEPT when authentication was not enabled.

**s03283: CBNUMBER parameter needed with callback operation
E164NUMBER**

An attempt was made to set the callback operation to E164NUMBER without a callback number being defined.

s03284: Incompatible parameters present

An attempt was made to set a global PPP parameter and an interface parameter as part of the same command.

s03285: Value missing on parameter PPP

An attempt was made to set a parameter that requires a specific PPP value but none was specified.

s03286: Parameter <string>required

A required parameter was not specified.

s03287: Can not destroy an interface when a user is attached

An attempt was made to destroy a PPP interface that has a user module attached. Delete the attached user instance and try again.

s03288: No LCPs are associated with this interface

The interface has no LCPs configured.

**s03289: Some interfaces and/or templates not destroyed because users
are attached**

Some of the PPP interfaces and/or templates were not destroyed during the purge because they had user modules attached. Delete the attached user instances and try again.

s03290: Value not allowed on parameter <string>

Some of the PPP interfaces were not destroyed during the purge because they had user modules attached. Delete the attached user instances and try again.

**s03294: PPP template <number> has users attached, so can not be
destroyed**

The given PPP template has users attached, so can not be destroyed. All users attached to this template have to be detached before the template can be destroyed.

s05256–s05999: Internet Protocol (IP)

s05257: Must specify <Parameter-List>

One or more compulsory parameters were not specified. Refer to the Reference Manual for information on the syntax of this command.

s05258: Operation not available for <Interface-Type>

The requested operation is not permitted on interfaces of the specified type.

s05259: Parameters <Parameter-List> required for <Interface-Type> interfaces

The specified parameters are required by the command for interfaces of the specified type.

s05260: Parameters <Parameter-List> invalid for <Interface-Type> interfaces

The specified parameter(s) may not be entered when the command is operating on that particular interface type. For example, the CIRCUIT parameter can not be specified when using ADD IP ARP on an Ethernet interface.

s05261: Specified IP interface does not exist

The specified interface does not exist.

s05262: Interface type not known or not supported by IP module

The type (ETH, PPP, etc.) of the interface parameter in the command is not known or not supported by the IP module. The IP module can not be attached to interfaces of that type.

s05263: Interface is already attached to IP module

The specified interface is already configured for use by the IP module.

s05264: <Object> already exists

The specified object (ARP, route, etc.) already exists.

s05265: Specified IP address and netmask values are inconsistent

The IP address and network mask values specified in the command are incompatible and inconsistent with each other.

s05266: <Object> not found

The specified object (ARP, route, etc.) can not be found.

s05267: The IP module is not enabled

The IP module is not active. The command can not be completed.

s05268: No matching <Object> found

The object (ARP, route, etc.) can not be found.

s05271: Internal Error: <Error-Description>

An internal error has occurred. Please report this to your distributor or reseller.

s05272: <Object> successfully deleted

The deletion operation was successful. The specified object has been deleted.

s05273: No more <Object> may be added

No more objects of that type may be added, since the internal table holding them is now full.

s05274: Metrics from host <ipadd> will now be increased by <number>

The ADD IP BOOST operation was successful. Metrics from the named host will be artificially increased by the specified amount.

s05275: <Object> successfully added

The object (ARP, route, route exclusion, etc.) has been successfully added.

s05280: An unexpected problem with <Configuration-Table> required it to be reset

The specified configuration table could not be updated for some reason, most likely because it was missing. The table was automatically recreated and the configuration update continued. However, some older configuration information may have been lost.

s05281: Source-routed IP packets will now be <string>

Any IP packets received with the source-route option will now be either forwarded or discarded, as stated in the message.

s05282: <Object> successfully updated

The named item was successfully updated.

s05283: <Function> is already enabled

The function specified is already enabled.

s05284: <Function> is already disabled

The function specified is already disabled.

s05285: IP address <ipadd> already assigned to another interface

The specified IP address is already in use on the interface. The same IP address may not be used on two interfaces.

s05286: Interface not attached to IP module

The specified interface is not configured for use by the IP module.

s05287: <Function> has been enabled

The specified option or feature has been successfully enabled.

s05288: <Function> has been disabled

The specified option or feature has been disabled.

s05289: IP module configuration has been reset

The configuration for the IP module has been reset and the module has been restarted to make that configuration active.

s05290: IP module has been reset

The IP module has been reset. Any active TCP connections have been disconnected.

s05291: <Function> not supported on <Interface-Type> interfaces

The selected function is not supported on that kind of interface.

s05292: There are no packets in the IP debug queue

There are no packets in the IP debug queue.

s05293: Network <ipadd> already used on another interface

The specified IP network is already in use on another interface of this router. Each IP interface on this router must be allocated a unique IP address in different IP sub-networks. No two interfaces on the same router may have the same IP address or be in the same network as any other IP interface.

s05294: Invalid IP address <ipadd>

The specified IP address is not valid.

s05295: Operation restricted to static ARPs only

Only static ARP entries may be modified or deleted by the user.

s05296: IP address <ipadd> already used on inactive interface <value>

The specified IP address is already in use on the named inactive interface. An interface is inactive if it references a lower-level interface (such as Ethernet, PPP, etc.) that no longer exists or could not be configured for use by IP. If this interface is no longer required it should be deleted. Otherwise you must assign a different IP address (in a different subnet) to this new interface.

s05297: Inactive interface deleted

The specified interface, which is inactive (normally because it could not be attached to the lower-layer interface) has been deleted.

s05298: Invalid parameters combination. SOURCE and ACTION/POLICY/PRIORITY required, DESTINATION, SMASK, DMASK, ENTRY, SPORT, DPORT, PROTOCOL are optional

An invalid combination of parameters was entered. Follow the command syntax as suggested in the error message, or in the Reference Manual.

s05299: Invalid parameters combination. FILTER and ENTRY required

An invalid combination of parameters was entered. Follow the command syntax as suggested in the error message, or in the Reference Manual.

s05300: Invalid filter number, must be in the range [0..299]

The specified filter number is invalid. Re-enter the command with a filter number between 0 and 99 inclusive.

s05301: The IP gateway is not currently active

The IP routing module is not currently active, so IP filtering will not operate. Enable the IP routing module first.

s05302: DESTINATION parameter must be defined for specified DMASK

The command entered included the DMASK parameter but not the DESTINATION parameter. If DMASK is present the DESTINATION parameter must also be specified.

s05303: Specified address-mask pair(s) are incompatible

An incompatible address/mask pair was specified. To be compatible, the condition ADDRESS & MASK = ADDRESS must be observed.

s05304: The specified filter has already been defined

The filter specified in the command has already been defined. Use the command SHOW IP FILTER to see the list of filters already defined.

s05305: Failed to add the specified pattern to filter list

The addition of the new filter to the filter list failed. It may be caused by low buffer space.

s05307: Specified pattern does not exist in the filter list

The filter with the specified pattern was not found. Use the command SHOW IP FILTER to see the list of all defined filter patterns.

s05308: Failed to delete the specified pattern from the filter list

The filter pattern could not be deleted from the filter list.

s05309: Specified filter is empty

The specified filter list is empty. Use the command SHOW IP FILTER to see the list of patterns defined for filters.

s05310: Protocol type must be specified for the given port

When adding a filter, if the source or destination filter is not "ANY" then a protocol type must be specified. Refer to the Reference Manual for valid protocol types.

s05311: For IP protocols other than TCP and UDP port numbers must not be specified

Only TCP and UDP protocols regard the port numbers as significant. For other protocols, port numbers are irrelevant.

s05312: Specified filter does not exist in the filter list

The specified filter does not exist. Use the command SHOW IP FILTER to see the list of currently defined filters.

s05313: Filter <number> has not been defined

The specified filter was not found in the filter list. Use the command SHOW IP FILTER to see the list of all currently defined patterns.

s05314: SESSION must not be specified for protocols other than TCP

The SESSION parameter is only valid for the TCP protocol (PROTOCOL=TCP).

s05315: IP address <ipadd> not valid as local address

The given IP address can not be used as a local IP address because none of the router's interfaces have this address. Re-enter the command specifying a valid interface address, or define an interface with the address required.

s05316: Specified ICMP type is incompatible with the ICMP code

The specified ICMP type is incompatible with the given ICMP code.

s05317: <string> and <string> parameters are only valid for ICMP protocol filter

User specified ICMPTYPE and/or ICMPCODE keywords for non-ICMP filter.

s05318: Inappropriate filter number for <string> filter

Wrong filter number for a filter type.

s05319: IP route already exists or not allowed by route filter

Invalid logical interface number.

s05320: Invalid IP logical interface number, must be between 0 - <number>

The specified logical interface number was invalid because it is outside the legal range.

s05321: No route to specified host

This router has no route to the specified IP address.

s05322: Unable to resolve host name to IP address

The specified host name could not be converted into an IP address. This normally means that the host does not exist.

s05323: No nameserver is defined, unable to perform hostname lookups

No name server IP address is defined. The router can not perform DNS queries to find the IP address of a named host.

s05324: Nameserver not responding

The router has received no response to DNS queries sent to the defined nameserver IP address.

s05325: Outbound Telnet connections not permitted

The user issuing the TELNET command does not have permission to initiate Telnet sessions.

s05326: Invalid IP address or host name: <string>

The argument to the PING command could not be interpreted as a valid IP address or a known host name (as defined in the IP host table).

s05327: Resolving host name "<string>" to IP address

A request has been sent to the Domain Name Server to resolve the specified host name to an IP address.

s05328: Host name resolved to <ipadd>

A response has been received from a Domain Name Server giving the required IP address. A Telnet connection to this address will be established.

s05329: Attempting Telnet connection to <Remote-Node>, Please wait...

A Telnet connection is being made to the specified remote node.

s05335: RIP entry for router's own interface is not allowed

A static RIP entry can not be added with an IP address that matches the IP address of one of the router's interfaces.

s05336: IP Route <string> is <string>

The user entered the command SHOW IP ROUTE COUNT but IP route counting is disabled.

s05337: Password too long, maximum 15 characters

The password is too long. The maximum password length is 15 characters.

s05338: Port value is invalid

The PORT value is invalid. Either it is 0, or greater than 65535, or is not a known named protocol type.

s05339: Invalid parameter combination

An invalid combination of parameters has been used.

s05340: static ENAT GBLIPADDR can not be lower than existing global range

An invalid combination of parameters has been used.

s05341: The given IP range is too large, maximum <number> addresses

The given IP address range contained too many entries. Try again with a smaller range of IP addresses.

s05342: The IP range overlaps with an existing range in <string>

The given IP address range overlaps with a previously defined range.

s05343: The <string> still contains active entries

The given item still has active entries and can not be altered.

s14256–s14999: Q.931

s14256: Unrecognised debug type, <string>

The Q.931 debugging type given is not recognised. Re-enter the command with a recognised value for the DEBUG parameter.

s14257: Already debugging this interface from another device

The interface for which debugging was requested is already being debugged and the information being sent to another device. Re-enter the command for a different interface or turn off debugging on the interface required.

s14258: Call not found

The call specified in the command was not found. Re-enter the command with a valid call number.

s14259: Non-existent Q.931 interface

The Q.931 interface specified in the command does not exist. Re-enter the command with a valid Q.931 interface number.

s14264: No Q.931 interfaces to display

There were no Q.931 interfaces to display. This router does not have ISDN interfaces present, so the command entered is meaningless.

s14265: No Q.931 calls to display

There are no Q.931 calls defined, so no calls can be displayed.

s14266: Invalid ISDN interface

The ISDN interface entered was invalid. Re-enter the command with a valid ISDN interface name.

s14267: Interface is allocated to TDM

The interface specified is not actually allocated to ISDN, and therefore can not be used for a Q931 command.

s14268: Already doing trace debugging on this device

Trace debugging is already taking place from this device. The command as entered is redundant.

s14269: Already doing trace debugging on another device

Trace debugging is already taking place from another device. Trace debugging can only take place from a single device at a time.

s14270: Message must be an even number of HEX digits

A Q.931 message has been specified which is an odd number of HEX digits. Since each octet of the message is represented by 2 HEX digits, this represents a non-integral number of octets, which is meaningless. Re-enter the command with an even number of HEX digits in the message.

s14271: Error parsing ASPID indices

An error was detected parsing ASPID indices. Indices must be decimal numbers separated by commas. There may be no more indices entered than there are valid DLCs in the router. The number of valid DLCs in the router is currently two.

s14272: Not waiting for manual selection of auto SPIDs

The ENABLE Q931 ASPID command was entered for an interface which is not running the auto SPID procedure and is not waiting for manual selection of one or more auto SPIDs. The wrong interface may have been selected, or the command is not required. It is possible to force auto SPID procedures to take place by entering the ACTIVATE Q931 ASPID command.

s14273: Auto SPID index is not in list

The auto SPID index specified in the command is not in the list of saved auto SPIDs. Re-enter the command and specify auto SPID indices which are in the list.

s18256–s18999: TEST Module**s18256: Invalid option "<string>"**

The string displayed in quotes is not a valid option of this command. Re-enter the command with the correct option.

s18257: Syntax error

The structure of the entered command could not be resolved. Re-enter the command correctly.

s18258: Missing parameter

This command requires another parameter before it can be executed. Re-enter the command correctly.

s18259: Missing option

This command requires another option before it can be executed. Re-enter the command correctly.

s18260: Test(s) halted

One or more tests have been disabled in response to a user command. The results of the test(s) can now be viewed.

s18261: No tests running, no action taken

The tests could not be disabled, since no tests are currently running.

s18262: Invalid parameter "<string>"

The string displayed in quotes is not a valid parameter of this command. Re-enter the command with the correct parameter.

s18263: Tests still running

This command could not be carried out because there are tests currently running. Disable the tests and then re-enter the command.

s18264: Interface test results cleared

The test results table has been cleared by the user command. The table is now in its default state, cleared of any previous test results.

s18265: Interface not specified

This command requires the interface to be specified. Re-enter the command specifying the required interface.

s18266: Requested test(s) already running, no action taken

The requested test was already enabled, so it was left to continue operation. If a restart of the test is required, first disable the test, then re-enter this command.

s18267: Invalid type "<string>"

The string displayed in quotes is not a valid test type. Re-enter the command specifying a valid test type.

s18268: Missing interface option

The interface name is missing from the command.

s18269: Interface not found "<string>"

The name displayed in quotes could not be found in the test table. Re-enter the command with the correct name.

s18270: Time out of range

The time entered for the length of the test is out of range. Re-enter the time with a range from 0 to 99000 minutes.

s18271: <string><number> is already assigned

The interface on which a test was requested is already attached for use by another user module. Change the configuration of the user module to detach the interface then re-enter this command.

s18272: BASIC tests are not implemented

The system of basic tests have not been implemented - only full tests are available.

s18273: Test on <string><number> halted; active LAN detected

The Ethernet test has been disabled due to the detection of an active LAN. This means that during the test external traffic was found on the LAN and the test was halted.

s18274: Warning test on <string><number>; data signal failure or transceiver not connected

A number of consecutive data frames sent out an Ethernet interface have not been detected as being looped back. The cause can be a missing or incorrect loopback; BNC requires a terminated segment, twisted pair a simple loopback and AUJ an AUJ loopback or terminated transceiver. Insert the correct loopback and re-enter the command. This message can also be caused by a device failure on the board being tested.

s18276: Test on <string><number> halted; data signal failure or loopback not connected

A number of consecutive data frames sent out an interface have not been detected as being looped back. On asynchronous ports and basic rate ISDN interfaces this can be caused by a missing or incorrectly wired loopback plug. Insert the correct loopback plug and re-enter the command. This message can also be caused by a device failure on the board being tested.

s18277: Test on <string><number> halted; coprocessor engine test failed

A number of consecutive data frames sent to the coprocessor engine have not been detected as being looped back. This can be caused by a device failure on the board being tested.

s18278: Test on <string><number> complete

The indicated test has been running for the specified length of time and has completed.

s19256–s19999: LAPD

s19256: Instance value required

This command requires a value for the LAPD instance. Use the SHOW LAPD command to identify the instance and re-enter the command specifying the instance.

s19257: Instance invalid; "<string>"

The string shown in quotes is not a valid LAPD instance. Re-enter the command using the correct instance value.

s19258: Instance not found

The specified instance could not be found in the LAPD instance table.

s19259: Parameter value required

The specified command is missing a parameter value. Re-enter the command with the value included.

s19260: Invalid parameter "<string>"

The string shown in quotes is not a valid parameter for this command.

s19262: Invalid option "<string>"

The string shown in quotes is not a valid option for this command.

s19264: SAP not found

The specified Service Access Point identifier does not exist in the LAPD table.

s19265: k parameter out of range

The value entered for the k parameter is outside the allowed range.

s19266: SAPI out of range

The value entered for the SAPI parameter is outside the allowed range.

s19267: lapdAttach - <string>

This is a debug message.

s19268: Attached to <string><number> SAPI <number>

This is a debug message.

s19269: lapdConnectRequest - <string>

This is a debug message.

s19270: Connection added to <string><number> SAPI <number> CES=<number>

This is a debug message.

s19271: CES required

This command requires a Connection Endpoint Suffix to be specified.

s19272: lapdEstablishRequest - <string>

This is a debug message.

s19273: Establish event made; <string><number> SAPI <number> CES <number>

This is a debug message.

s19274: lapdReleaseRequest - <string>

This is a debug message.

s19275: Release event made; <string><number> SAPI <number> CES <number>

This is a debug message.

s19276: lapdDataRequest - <string>

This is a debug message.

s19277: I frame event created; <string><number> SAPI <number> CES <number>

This is a debug message.

s19278: lapdDataRequest repeat <number> - <string>

This is a debug message.

s19279: <number> I frame events created; <string><number> SAPI <number> CES <number>

This is a debug message.

s19280: lapdUnitDataRequest - <string>

This is a debug message.

s19281: UI frame created; <string><number> SAPI <number> CES <number>

This is a debug message.

s19282: lapdUnitDataRequest on repeat <number> - <string>

This is a debug message.

s19283: CES required

This command requires a Connection Endpoint Suffix to be specified.

s19284: DLC not found

The requested Data Link Connection does not exist.

s19285: LAPD test set

This is a test message.

s19286: Test SAP and CES not set

This is a test message.

s19287: Test value out of range

This is a test message.

s19288: Case value out of range

This is a test message.

s19290: Can not add TEI in automatic TEI mode

A TEI can not be added to a LAPD interface which is in automatic TEI mode. If the TEI must be set change the interface mode to non-automatic and re-enter the command.

s19291: TEI out of range

The entered TEI is out of range. Re-enter the command using a valid TEI value.

s19292: TEI already exists

The specified TEI already exists. Delete the existing TEI first or re-enter the command using a new TEI value.

s19293: Only one TEI allowed for NT

In NT mode only a single TEI may be used. Delete the existing TEI before adding another.

s19294: TEI list full

The LAPD TEI list is full. Delete one of the existing TEIs before adding another one.

s19296: TEI does not exist

The specified TEI does not exist on this interface.

s19297: SAP <number> deleted

The displayed SAP has been deleted and any user module attempts to use this SAP will now fail.

s19298: CES not found

The specified CES does not exist.

s19299: CES <number> of SAP <number> deleted

The displayed CES has been deleted and any user module attempts to use this CES will now fail.

s19300: Too many timers!

The LAPD module has exceeded its maximum timer queue length. Contact your distributor or reseller for assistance if this is preventing correct operation of the ISDN.

s19302: XTEI or XSPID out of range

The entered XTEI or XSPID is out of range. Re-enter the command using a valid XTEI value (0-63) or a valid XSPID value (0-1).

s19303: XTEI or XSPID already exists

The specified XTEI or XSPID already exists. Delete the existing XTEI or XSPID first or re-enter the command using a new XTEI or XSPID value.

s19304: XTEI or XSPID list full

The LAPD XTEI or XSPID list is full. Delete one of the existing XTEIs or XSPIDs before adding another one.

s19306: XTEI or XSPID does not exist

The specified XTEI or XSPID does not exist on this interface.

s22256–s22999: TCP**s22256: The given index does not reference a current TCP session**

The user supplied index does not reference a current TCP session

s22257: You can't delete a TCP session in listen state

The user has tried to delete a TCP session in the listen state. This is not allowed

s23256–s23999: Ethernet Driver**s23256: There are no Ethernet instances**

The router has no Ethernet interfaces. There is no point in entering Ethernet commands.

s23257: Ethernet instance <Interface-number> does not exist

The Ethernet interface number specified does not exist. Re-enter the command with the instance number of an existing Ethernet interface.

s23258: Ethernet instance has been RESET

The Ethernet interface specified has been reset. Hardware is reset and transmit and receive queues are purged.

s23259: Ethernet instance must be specified

A Ethernet interface number "n" must be specified for this command as ETH=n. Re-enter the command specifying an Ethernet instance.

s28256–s28999: Compression**s28256: No engine found**

No coprocessing engine has been found. If an engine is installed power the router off, check that it is plugged in correctly, power on the router and then re-enter the command.

s28257: Mutually exclusive parameters on same line - <string> and <string>

In parsing a command, two mutually exclusive parameters were seen. Re-enter the command, using only one of these parameters.

s28258: Parameter not seen. <string> must precede <string>

Another parameter must appear before one of the parameters entered.

s28259: Invalid address: not 4-byte aligned

The address value must be a multiple of four.

s28260: Engine operation failed

The engine operation was unsuccessful.

s28261: Entered key was corrupted

Valid keys contain a checksum, but the checksum of the key entered was incorrect.

s28262: Parameter value not seen. <string>=value must precede <string>

The two parameters must be specified in the correct order.

s28263: ENCO access enabled

ENCO access has been enabled.

s28264: Password incorrect, access denied

The entered password did not match the one stored on the system.

s28265: ENCO access disabled

ENCO access has been disabled.

s28266: ENCO access already enabled

ENCO access is already enabled. The command entered had no effect.

s28267: Password too short - must be at least <number> characters long

The password entered is shorter than the minimum length required.

s28268: Invalid port - command not executed

ENCO access controlled commands must be entered via port 0.

s28269: Enable ENCO access retry within 10 seconds of failed attempt

Ten seconds must elapse between ENCO access enable retries.

s28270: New ENCO access password. Access disabled

A new password has been entered and stored. Access has been disabled and will have to be re-enabled with the new password.

s28271: Command not successful - password not changed

The new password was not successfully stored. The old password is still effective.

s28272: <string> value length too short. Minimum length is <number>

The specified parameter's value must be at least the specified length.

s28273: ENCO access not enabled - command not executed

The authorization check failed so the command was not executed.

s28274: Value not allowed on parameter <string>

In the entered command the specified parameter can not take a value.

s28275: Slave serial number <string> not in log

No slave with the serial number entered has requested the master key table.

s28276: Transfer already authorised and proceeding

The specified request has already been authorised and the transfer is underway.

s28277: No current request from specified slave

The specified request has failed or expired.

s28278: No valid ENCO access password exists - Key Memory unsecured

No writes to the engine key memory are allowed if it is not protected by a valid ENCO access password.

s28282: A new boot configuration script must be created and the router rebooted before the number of software channels that can use Stac LZS compression will change

The number of Stac LZS channels has been changed and a reboot is required.

s28283: Can not use software compression when an engine is present

Can't set software compression parameters when there is an engine present

s28284: Not enough buffers set aside for software compression

Insufficient buffers were reserved at startup for the requested number of channels.

s28285: Buffers set aside for software compression are not contiguous

The buffers reserved at startup for software compression are not contiguous.

s30256–s30999: X.25 Layer 3 (DTE)**s30256: DTE interface specified does not exist**

The specified DTE interface does not exist. Use the command SHOW X25T[=x25-interface] to display the list of DTE interfaces defined.

s30257: No value allowed for parameter X25T

The X25T parameter in the specified command does not accept a value.

s30258: Call parameter instance number <number> already exist

When adding a call parameter instance, a call parameter instance with the same number has been found in the call parameter table. Use the command `SHOW X25T CPAR[=call-index]` to display the list of currently defined call parameter instances.

s30259: Call parameter <number> does not exist

The specified call parameter instance does not exist. Use the command `SHOW X25T CPAR[=call-index]` to display the list of currently defined call parameter instances.

s30260: <string>, string not even length

The parameter `USERDATA` requires a value that is a data string of even length. Re-enter the command again with a user data string which has an even length.

s30261: Interface must be specified

The X25T interface number must be specified when entering the command.

s30262: PVC index must be between 1 and <number>

An invalid PVC index was specified.

s30263: Specified PVC already exists

The specified PVC to be added already exists in the PVC table. Use the command `SHOW X25T[=x25-interface] CIRCUIT` to display the list of all or a given circuit's PVCs.

s30267: Invalid parameters combination

An invalid combination of parameters was entered. Refer to the Reference Manual for the correct command syntax.

s30268: DTE instance number <number> already exists

The specified DTE instance already exists. Use the command `SHOW X25T[=x25-interface]` to display the list of DTE interfaces defined.

s30269: <string>, invalid parameter value

An invalid value was entered for the specified parameter.

s30270: Unsupported link layer implementation

The specified link layer entity is not currently supported. Refer to the Reference Manual for a list of currently supported LLEs for X25T interfaces.

s30271: LLE instance specified is invalid or does not exist

The specified LLE instance does not exist or is invalid. Use the `SHOW` command for the appropriate LLE module to see the currently defined LLE instances.

s30273: Channel parameters are not consistent

One or more values entered for the `LIC`, `HIC`, `LTC`, `HTC`, `LOC`, `HOC`, `NPVC` and `MAXACTIVE` parameters are inconsistent with the others. Refer to the Reference Manual for consistency rules for these parameters.

s30279: <string>, invalid parameter

An invalid parameter was entered in the command line. Refer to the Reference Manual for the correct command syntax.

s30281: Default call parameter <number> does not exist

The specified default call parameter does not exist. Use the command `SHOW X25T CPAR` to display the list of all call parameter instances defined.

s30282: <string> and <string> must both be 0 or both be greater than 0

One of the given channel range limits is zero, which means the limit is not currently in use. Both of the channel range limits given must be zero (i.e. there are no channels of that type usable) or greater than zero (i.e. channels of the indicated type are available within the given range).

s30283: <string> (<number>) must be greater than <string> (<number>)

The channel ranges specified are overlapping.

s30284: <string> (<number>) must be the same as <string> (<number>) or greater than it

An inconsistent channel range has been specified. The highest limit must be the same as or greater than the lowest limit.

s30285: Message must be an even number of HEX digits

A X25T message has been specified which is an odd number of HEX digits. Since each octet of the message is represented by 2 HEX digits, this represents a non-integral number of octets, which is meaningless. Re-enter the command with an even number of HEX digits in the message.

s31256–s31999: FLASH Driver

**s31256: Specified file name is invalid. Must be in the form of
DDDD:MMMM\NNNNNNNN.TTT**

The file name entered had an invalid syntax. The correct syntax for a file name is DDDD:MMMM\NNNNNNNN.TTT, where DDDD is the device name (e.g. FLASH), MMMM is the module name, NNNNNNNN is the file name and TTT is the extension.

s31257: Invalid data byte <<string>>. Must be in the range of [00..FF]

An invalid data string was entered. The data string must consist of hexadecimal numbers (0x00 - 0xFF), and must be even length string.

s31258: Invalid hexadecimal data byte: <<string>>

An invalid data string was entered. The data string contains an invalid hexadecimal number. A valid data string must consist of hexadecimal numbers (0x00 - 0xFF), and must be of even length.

**s31259: Invalid parameter combination. The correct command should be:
<string>**

An invalid parameter combination was specified.

s31260: Flash compacting... DO NOT restart the router until compaction is completed

The message tells the manager not to restart the router while a flash compaction is under way, otherwise flash could become corrupt.

s31261: Flash compaction successfully completed

The message tells the manager that the flash compaction has successfully completed.

s31262: Flash compaction failed. Error = <number>

The message tells the manager that the flash compaction has failed.

s31263: Flash file rename underway... DO NOT restart the router or alter Flash until rename is completed

The message tells the manager not to restart the router while a flash file rename is under way, otherwise flash could become corrupt.

s31264: Flash file rename successfully completed

The message tells the manager that the flash file rename has successfully completed.

s31265: Flash file rename failed. Error = <number>

The message tells the manager that the flash file rename has failed.

s33256–s33999: TELNET**s33256: Attempting Telnet connection to <Remote-Node>, Please wait...**

A Telnet connection is being made to the specified remote node.

s33257: The TELNET termtype is set to: <string>

The message shows the current setting of the telnet termtype.

s33258: Outbound Telnet connections not permitted

The user issuing the TELNET command does not have permission to initiate Telnet sessions.

s33259: There are no free sessions available

The maximum number of sessions is exceeded. Delete one or more sessions before proceeding to create another session.

s34256–s34999: System**s34256: Insufficient memory to allocate help tree**

The message shows that the router has insufficient memory to allocate another help structure.

s34257: The help file is corrupt

The message shows that the router is unable to successfully read the help file, due to corruption.

s34258: The help file <string> was not found on the system. No online help is currently available

The specified help file no longer exists on the system.

s34259: No help is available on the system. Please load the appropriate help file onto the system, or refer to the router reference manual

No file could be found. No help file is specified and no default help file could be found.

s34260: The help file was not completely read

Due to a file error the help file was not totally read.

s34261: Unknown help command

The help command was not parsed to a known help page

s34262: Flash is currently busy, try the router restart later

The FLASH is currently busy, and restarting the router at this time will may corrupt the files stored in FLASH.

s34263: Q931 and PBX parameters (where applicable) set to defaults for specified territory

Setting the territory causes various territory related parameters in the Q931 and PBX modules are to be set to their default values for that territory. Some or all of these modules may not be applicable to a particular router.

s35256–s35999: Command Processor**s35256: Unknown command "<string>"**

An unknown command was detected. Either the command is not supported or there was a typing error entering the command. Refer to the Reference Manual for the correct command syntax and re-enter the command.

s35257: Privilege violation

A privilege violation was detected in processing the command. The command can not be successfully executed unless the privilege is changed to the correct level. Re-enter the command after the privilege level has been changed.

s36256–s36999: TTY**s36256: Specified TTY <string> is too long**

The specified TTY prompt is too long. Refer to the Reference Manual for the maximum prompt length.

s36257: <string> Command: SET TTY [HISTORY=history] [PAGE=page] [PROMPT=prompt] [TYPE=type]

When entering the command, an invalid combination of parameters was specified. Follow the command syntax as suggested in the error message.

s36258: Specified TTY instance is not configured

The specified TTY instance is not configured. Use the command SHOW TTY[=ALL] to display the list of all TTY instances configured.

s36259: <string> Command: SHOW TTY [{={ALL|n}}][SUMMARY]

When entering the command, an invalid combination of parameters was specified. Follow the command syntax as suggested in the error message.

s36269: Not enough privilege to <string>, must be MANAGER privilege

There is not enough privilege to execute the command. Log in as "MANAGER" or as a user with manager privilege and re-enter the command.

s36277: Command number not found

The command history buffer does not contain a command with the specified number. Use the command SHOW PORT HISTORY to display the command history.

s36278: SHOW PORT HISTORY the history list is empty

The command history buffer is empty.

s36279: Not permitted to show tty(s), unless with MANAGER privilege

The specified command requires MANAGER privilege. Log in as "MANAGER" or as user with manager privilege, and re-enter the command.

s36280: User is not logged on

The specified user is not logged on to the system.

s36281: Connect fails because of security reason

The connection has been refused because of security problems. Refer to the Reference Manual for more information.

s36283: Invalid port number

The specified port number is invalid.

s36284: The filename given is invalid

The given filename is not valid for some reason, such as being too long or having a bad file type.

s36285: This file is not a text file and can't be edited

The given file type can't be edited. This message is displayed if a user attempts to edit a RELEASE file.

s36286: The terminal type for this tty is set to dumb

The terminal type for this TTY is set to DUMB. The TTY terminal type should be set to VT100 before an editing session can start.

s36287: Syntax: EDIT [filename]

The correct syntax for the edit command is "EDIT [filename]".

s36288: Flash is currently busy, try again later

The FLASH memory is busy (probably compacting). Don't try editing files at this time.

s36289: Insufficient storage space to edit this file

There is not enough storage space to edit this file

s37256–s37999: ISDN Call Control**s37256: Parameter <string>, invalid ISDN call name**

A parameter that should be an ISDN call name had an invalid value. Re-enter the command with a valid ISDN call name.

s37257: Call name can not be a number

The call name entered consisted entirely of decimal digits and so could be interpreted as a decimal number. This is not allowed for call names. Re-enter the command with a valid call name.

s37258: Call name too short

The call name entered was too short. Re-enter the command with a name of the correct length range.

s37259: Call name too long

The call name entered was too long. Re-enter the command with a call name in the valid length range.

s37260: Active call not found

The active call specified was not found. Re-enter the command specifying a call index that is actually active.

s37261: Invalid ISDN interface

The ISDN interface entered was not well-formed. Re-enter the command specifying an ISDN interface with valid syntax.

s37262: ISDN interface not found

The ISDN interface specified does not exist on this router. Re-enter the command specifying an interface that does exist.

s37263: Call already exists

The call specified in the command already exists and can not be added again. Re-enter the call with a new name, or use the SET command to change the existing call name.

s37264: Call not found

The call specified was not found and can not be modified. Re-enter the command with a call that exists, or use the ADD command to add a new call.

s37265: Call <string> has been activated as active call <number>

The specified call has been activated, with the given active call index.

s37266: Call has no user attached

The specified call could not be activated because no higher layer module, such as PPP, is attached to the call. Make sure that there is a higher layer attached to the call before activating.

s37267: No free user instances

The call activated had no free instances available for making this call. Another instance of the higher layer over this call will have to be created before reactivating the call.

s37268: No resources for the call

A resource such as memory, a B channel or an active call slot is missing and the call can not be activated. Check that call slots, B channels and memory are available and reactivate the call.

s37269: Call is disabled

The call specified for this command is disabled. Either enable the call and re-enter the command or use a call that is enabled.

s37270: Invalid return code <number>

The return code from an internal function is invalid and has the stated value. Report the error message to your distributor or reseller, along with the command that was entered.

s37271: Parameter <string> missing

The parameter specified is missing from the command line. Re-enter the command and supply the missing parameter.

s37272: Call name the same as remote call name

The call name and remote call name can not be the same since they are used to determine the precedence of the call, and if they are the same this can not be done. Re-enter the command, specifying a different call name for the call or remote call.

s37273: No more calls allowed

There is a limit to the number of call details allowed, and this limit has been reached. The call can not be added unless another call is deleted.

s37275: All active calls for call <string> deactivated

All active calls for the given call have been deactivated.

s37276: One active call deactivated

One active call was deactivated.

s37277: Call has attachments

The call in the command has higher layer modules attached to it, so can not be deleted. Clear the higher layer attachments before attempting to delete the call.

s37278: Call already disabled

The specified call is already disabled. Re-enter the command with a call that is enabled.

s37279: Call already enabled

The specified call is already enabled. Re-enter the command with a call that is disabled.

s37280: ISDN log already enabled

The ISDN call log is already enabled.

s37281: ISDN log already disabled

The ISDN call log is already disabled.

s37282: ISDN call logging is disabled

The ISDN call logging function is disabled, so there is nothing to show in the log. Enable ISDN call logging to get some output from the SHOW ISDN LOG command.

s37283: No calls in ISDN log

There are currently no calls in the ISDN log.

s37284: No ISDN call details found

There were no ISDN call details found on this router.

s37285: No ISDN active calls found

There were no ISDN active calls found on this router.

s37286: Number <string> already exists on CLI list <number>

The specified number already exists on the specified CLI list. Re-enter the command specifying a new CLI list or a different number.

s37288: Number <string> not found in CLI list <number>

The specified number was not found in the specified CLI list. Re-enter the command specifying the correct list or the correct number.

s37289: CLI list <number> is empty

The specified CLI list is empty and can not be displayed. This is an informational message.

s37290: There are no CLI lists defined

In a request to show all CLI lists, nothing can be displayed since there are no CLI lists defined. This is an informational message.

s37291: Interface not allocated to ISDN

The interface specified is not actually allocated to ISDN, and therefore can not be used for an ISDN command.

s37292: The default domain name is <string>

The domain name displayed is appended to the login name for a Domain Name Service (DNS) query in order to find an Internet Protocol (IP) number to be associated with the user. This is only done if the user name supplied does not contain a "." (which is taken to indicate that the name is already fully qualified), and a default domain name has been specified.

s38256–s38999: MIOX

s38256: <Interface>, specified interface does not exist

The specified interface does not exist.

s38257: Invalid parameter combination

The combination of parameters specified was invalid.

s38258: Unknown internal error

While processing the command, an unexpected internal error occurred. This normally indicates a software problem or a corrupt configuration. Please contact your distributor or reseller for assistance.

s38259: No circuits configured for interface

No circuits have been configured for the specified interface.

s38260: Specified circuit not found for interface

The specified circuit could not be found.

s38261: Specified circuit not found for any interfaces

The specified circuit could not be found.

s38262: No circuits configured for any interfaces

No circuits have been configured.

s38263: No interfaces configured

No interfaces have been configured.

s38265: <Circuit>, specified circuit name does not exist

The specified circuit does not exist.

s38266: <DTE-address>, specified circuit DTE address already exists

The DTE address specified already exists.

s38267: <Circuit>, circuit is unavailable or does not exist

The specified circuit is not available for use or does not exist.

s38268: <CPAR-entry>, specified call parameter entry does not exist

The specified call parameter entry does not exist.

s38269: Must not specify both DTEADDRESS and PVC

It is not valid to specify both the DTEADDRESS and PVC parameters for this command.

s38270: PVC circuits can not support multiple encapsulations

PVC circuits are not able to support multiple encapsulations.

s38271: Encapsulation does not support attached user modules

The encapsulation is singular and there is more than one module attached or the attached module is not supported.

s38272: <Circuit>, specified circuit name already exists

The specified circuit name already exists.

s38273: Must specify CIRCUIT circuit name

A circuit name is required but was not entered.

s38274: Must specify either DTEADDRESS or PVC

At least one of the DTEADDRESS and PVC parameters must be supplied.

s38275: Missing parameter

A parameter is missing for the specified command.

s38276: Circuit is still in use

When deleting a circuit, a check is made that no user module is using the circuit before it can be deleted.

s38277: Circuit is already enabled

The circuit is already enabled.

s38278: Circuit is already disabled

The circuit is already disabled.

s38279: Can not activate a PVC circuit

The activation of a circuit failed because of an internal error.

s38280: Must specify parameter USER for multiple encapsulation

The USER parameter must be specified when the encapsulation type of the circuit is "MULTIPLE".

s38281: Circuit user module is not attached to circuit

No user module is configured to the specified MIOX circuit.

s38282: Circuit is disabled

When activating or deactivating a MIOX circuit, a check is made to see if the specified circuit is enabled.

s38283: Circuit call is already active

The circuit to be activated is already in the active state.

s38284: No available channels to activate call over

A call collision has occurred.

s38285: Call is already being attempted

The call is already open or is being setup.

s38286: Can not deactivate a PVC circuit

PVC channels are by default always active and therefore can not be deactivated.

s38287: Must not specify parameter USER for singular encapsulations

When a singular encapsulation is used, the USER parameter must not be specified.

s38288: Call failed as compression unavailable

The MIOX call failed because software compression is not enabled or hardware compression is unavailable.

s39256–s39999: BOOTP**s39257: Max. number of relay dest. addresses exceeded**

No more relay destination addresses may be defined.

s39258: Specified relay destination does not exist

The specified relay destination does not appear in the current configuration.

s39259: Specified relay destination already exists

The specified relay destination is already defined.

s39260: BOOTP relay agent is already disabled

The BOOTP agent is already disabled.

s39261: BOOTP relay agent is already enabled

The BOOTP agent is already enabled.

s39262: Unknown internal error

An internal error occurred while processing the command. This normally indicates a software problem or a corrupt configuration. Please contact your distributor or reseller for assistance.

s39263: Invalid BOOTP relay destination

The specified BOOTP relay destination is invalid.

s41256–s41999: BRI Driver**s41256: Instance has been RESET**

The BRI interface specified has been reset. The hardware is reset and the S/T loop activation procedure is reinitialised. If a frame is being transmitted the transmission is aborted. The receive queue is purged.

s41257: Instance must be specified

A BRI interface number "n" must be specified for this command as BRI=n.

s41258: Another test already enabled

Only one BRI CTEST may be enabled at a time. Use the command DISABLE BRI CTEST to disable the currently active CTEST before enabling another test.

s41259: Interface "<Interface-number>" not present

The specified BRI interface number does not exist.

s41260: The MODE parameter is required with the ISDNSLOTS or TDMSLOTS

The desired BRI mode must be specified if the ISDN slots or TDM slots for the interface are to be changed.

s41261: When MODE is ISDN the TDMSLOTS parameter is not allowed

When setting the BRI mode to ISDN it makes no sense to specify the TDM slots; only the ISDN slots may be specified.

s41262: When MODE is TDM the ISDNSLOTS parameter is not allowed

When setting the BRI mode to TDM it makes no sense to specify the ISDN slots; only the TDM slots may be specified.

s41263: When the MODE is MIXED both ISDNSLOTS and TDMSLOTS required

When setting the BRI mode to MIXED both the ISDN and TDM slot masks must be specified.

s41264: TDMSLOTS value is not compatible with current TDM groups

The change to the slots available to the TDM module for this interface is not permitted as it would remove some slots that are currently in use by TDM groups.

s41265: ISDNSLOTS value is not compatible with active ISDN calls

The change to the slots available for ISDN calls for this interface is not permitted as it would remove some slots that are currently in use by active ISDN calls.

s41266: Illegal slot specified, only slots 1 and 2 are available on BRI

An illegal slot list was specified for the ISDNSLOTS or TDMSLOTS parameter. The only slots available on a BRI interface are 1 and 2.

s41267: The same slot is included in ISDNSLOTS and TDMSLOTS

The slots available for ISDN calls may not include any of the slots specified for TDM groups and vice-versa.

s41268: U interfaces do not support semipermanent ISDN circuits

In the USA, semipermanent ISDN circuits are not available so the interface MODE may not be set to TDM or MIXED, nor may the ACTIVATION parameter be set to ALWAYS.

s43256–s43999: PORT Driver**s43256: PORT is the only permitted parameter**

PORT is the only permitted parameter for the ENABLE PORT or DISABLE PORT commands.

s43257: Illegal parameter. RESET PORT [{HISTORY|COUNTERS}]

HISTORY and COUNTERS are the only parameters permitted for the RESET PORT command.

s43258: Illegal parameter. SHOW PORT {SUMMARY|HISTORY} or SHOW PORT[={ALL|n}] or SHOW PORT[=n] COUNTERS={INT|RS232|DIAG}

An illegal combination of parameters was entered. The legal possibilities are SHOW PORT {SUMMARY|HISTORY}, SHOW PORT[={ALL|n}] and SHOW PORT[=n] COUNTERS={INT|RS232|DIAG}.

s43259: Not enough privilege to <string>

The user does not have the privilege required to enter the specified command. A user may not display the configuration of any port other than their own, nor may they display the port counters. A user at a port not set to "secure" may alter their own port settings, but may not if the port is secure.

s43260: Can not <string> own port

A user may not enable or disable their own port.

s43261: Not enough privilege to <string> port <number> (not own port)

The user does not have the privilege to issue the specified command. A user with user privilege may only issue SET commands for their own port and only if the port is not set to "secure".

s43262: Port is unused

This is an internal error. It indicates that there is no TTY attached to the port. Refer this problem to your distributor or reseller.

s43265: Port is already <string>

The port is already in the specified state.

s43266: Invalid port number

The port number specified is not available on the router. Use the command `SHOW PORT=ALL SUMMARY` to display the ports that are available. Specifying all ports is permitted with only some commands.

s43267: Cannot <string> virtual port

When a user Telnets to a router they are assigned a virtual port. A virtual port may not be reset. Only a small subset of `SET PORT` parameters may be used with virtual ports. These are `PROMPT`, `HISTORY`, `PAGE` and `TYPE`.

s43268: Invalid for host port

This is an internal error. It indicates that the TTY attached to the port is assigned. Refer this problem to your distributor or reseller.

s43270: Cannot loopback own port

A user may not put their own port into loopback mode.

s43271: Specified port <string> is too long

The string entered for the specified parameter is too long. The maximum string length for a port name, port prompt or service name is 15 characters.

s43276: Attention must be BREAK when speed is AUTO

If the port is set to determine the baud rate automatically then the attention character must be `BREAK`. This is so that the port may be forced to initiate a new search for the baud rate by pressing `BREAK`, which is independent of baud rate.

s43280: Use of the parameter IPADDRESS also requires the use of NETMASK

When the IP address for a port is being set, the network mask must also be set, and vice-versa.

s43281: Configuration updated

The `SET PORT` command was successfully actioned and the port configuration has been written to nonvolatile storage.

s43282: No ports fitted requirements

There are no ports on this router.

s43283: SHOW PORT=n is not permitted in USER mode, use SHOW PORT command for showing own port

A user without manager privilege may only display the configuration of their own port.

s43284: Too many parameters, use SET MANAGER PORT={n|NONE}

When setting a semipermanent manager port the only parameters permitted are `MANAGER` and `PORT`, with the port number or "`NONE`".

s43285: There is no semi-permanent manager port now

The semipermanent manager port has been reset to user privilege and will have user privilege the next time the router reboots.

s43286: Port <number> is the semi-permanent manager port

The specified port is now a semipermanent manager port. This means that if the router reboots the port's initial state will be as if the manager has logged into it.

s43287: Can't <string> for port other than asynchronous port

The commands SHOW PORT SUMMARY and SHOW PORT COUNTERS are invalid for a virtual port as they display information that is only applicable to asynchronous ports.

s43288: No user is logged in, can not enter security commands

The command SET MANAGER PORT is a security command and as such may only be issued when a manager has logged in. The manager may be challenged to re-enter their password if it is some time since s/he logged in.

s43289: Can not show history for all ports

It is not possible to display the history for all ports. Re-enter the command without specifying ALL, or simply press control C.

s43291: Page value out of range: <number> - <number>

The value entered for the page length is outside the valid range, which is displayed in the error message. Re-enter the command with a value within the range.

s43292: Specified speed not available for that port

Router asynchronous ports do not all support the same set of speeds. An invalid speed for the port has been specified.

s43293: PURGE PORT=*n*all, value must be given

The PURGE PORT command is used to restore one or more ports to their default values. Either the port number or ALL must be specified. Note that the current configuration of the port/s will be lost.

s45256–s45999: User Authentication Facility**s45256: <string>, command aborted for security reason**

The command was aborted because of security reasons. For example, an invalid password was entered or the command timed out.

s45257: User password should have length of <number> or more chars

The specified user password is of insufficient length. The minimum password length is specified by the MINPWDLEN parameter in the SHOW USER CONFIGURATION output.

s45258: <string>, USER parameter must not have value to show global USER module configurations

The USER parameter in the command SHOW USER CONFIGURATION was entered with a value attached to it. Re-enter the command without a value for the USER parameter.

s45259: <string>, USER parameter must not have value to set global USER module parameters

The USER parameter in the command SET USER [LOGINFAIL=1..10] [LOCKOUTPD=1..30000] [MANPWDFAIL1..5] [SECURDELAY=10..300] [MINPWDLEN=1..23] [TACTRETRIES=0..10] [TACTIMEOUT=1..60] was entered with a value attached to it. Re-enter the command without a value or the USER parameter.

s45260: <string>, user with that name already exists

When adding a user account, the specified user name has been found in the user database. Use a different user name for this user account.

s45261: <string>, password required

When adding a new user into the user database, the PASSWORD parameter must be specified.

s45262: <string>, both IPADDRESS and NETMASK are required

When adding a new user into the user database, if either the IPADDRESS or NETMASK parameter is seen then the other must also be specified.

s45263: Cannot delete last Manager privilege user when not in system secure mode

An attempt was made to delete the last remaining account with manager privilege, while the router is in system secure mode. This is not allowed because without this last account, there would be no account with the necessary privilege to manage the router.

s45264: <string>, user name not found in database

The command entered required an existing user name to be entered, but the specified user was not found in the user database.

s45265: <string>, user <string> has been deleted

The specified user has been deleted from the user database.

s45266: <string>, command must be in the form of: RESET USER[="string-15"] COUNTERS[={ALL|GLOBAL|USER}]

When entering a RESET USER command, an invalid parameter combination was detected. The correct syntax is RESET USER[=username] COUNTERS[=ALL|GLOBAL|USER].

s45267: <string>, if specific user is given COUNTER value must be USER

When entering a RESET USER command, an invalid parameter combination was detected. If the user name is specified, only the user counter can be reset.

s45268: <string>, specified counters have been reset

The specified counters have been reset.

s45269: <string>, user database has been purged

The user database has been purged or cleared.

s45270: <string>, must specify user login name to set user specific parameters

When setting a user specific parameter, a user name must be given.

s45272: <string>, no match was found to your query

When displaying a users account details, the specified user was not found in the user database.

s45273: <string>, user database is empty

The user database does not contain any user information.

s45274: This device is locked temporarily (login-lock)

The device is locked because too many login failures were detected.

s45275: <string>, the specified IP address was invalid

The given IP address was invalid. The IP address must be entered in the dotted decimal form (e.g. 202.25.44.12). The values "NONE" and "OFF" are also permitted.

s45282: SET PASSWORD, password being set by another user

When trying to set a password for a user, the router detects that some other user is trying to set a password as well. Try again after a short period of time.

s45283: No user is logged in, can not enter security commands

The security command can only be entered when a user is logged in. Try logging in with your user account before entering any security command.

s45284: LOGIN logout is activated

The LOGIN logout feature is activated for security reasons, such as too many incorrect passwords.

s45285: SET PASSWORD, no value allowed

The command SET PASSWORD must be entered without any value.

s45286: Invalid password for user

The specified password does not match the password stored in the database for the specified user.

s45287: SET PASSWORD, confirm password incorrect

When setting the password, the router will ask you for password confirmation. This message is generated when the confirmation password is not the same as the new password to set.

s45288: SET PASSWORD, internal error

An internal error was detected trying to set a user password. Report this problem to your distributor or reseller.

s45292: Invalid password was entered more than <number> times.**Logging off the connection now....**

The router detects that incorrect password has been entered more than the specified number of times. The connection to the router will be terminated.

s45293: MANAGER account may not be disabled

As management of the router depends on the availability of a manager privileged account, the MANAGER account may not be disabled.

s45294: MANAGER account may not have privilege set to USER

As management of the router depends on the availability of a manager privileged account, the MANAGER account may not have its privilege level set to user.

s45295: The username is invalid

The username given is invalid as it contains spaces.

s45296: <string> is already enabled

An attempt was made to enable a feature that was already enabled.

s45297: <string> is already disabled

An attempt was made to disable a feature that was already disabled.

s45298: No <string> configured

There were no objects of this type configured.

s45299: Invalid port - command not executed

USER access controlled commands must be entered via port 0.

s48256–s48999: LOADER

s48256: No Error

No error occurred. The operation was completed successfully.

s48257: TFTP server address required

The IP address of the TFTP server must be specified.

s48258: Load already in progress

The router is presently loading software from a remote server. Further load requests can not be initiated until this operation has completed.

s48259: The existing file must be deleted before overwriting

A file with the specified name already exists. If you wish to replace that file with another by the same name you should delete the original file first.

s48260: File not found by TFTP server

The TFTP server has reported that it can not find the requested file.

s48261: No response was received from the TFTP server

The TFTP server did not respond. Check the IP configuration, network connections and TFTP server configuration.

s48262: Insufficient memory space for file

The file is too large to be accommodated within the router's memory.

s48263: Invalid destination for release file

The destination specified for the file is invalid.

s48264: File header invalid

The file header was invalid. This file is probably corrupt.

s48265: File trailer invalid

The file trailer was invalid. This file is probably corrupt.

s48266: File contents invalid

The contents of the file appear to be invalid. The file is probably corrupt.

s48267: File not intended for this router model

The file is intended for another model of router, and can not be used on a router of this type. You should obtain a version of this file for this model of router, or consult your distributor or reseller for assistance.

s48268: Error in S record format

The contents of the file were not formatted correctly. The file is probably corrupt.

s48269: File load from TFTP server <ipadd> has begun

The router has established a connection to the TFTP server and has commenced the file transfer.

s48270: File transfer successfully completed

The file transfer has completed successfully.

s48271: Write to file failed

The router was unable to write the file to the requested destination.

s48272: File transfer failed with TFTP error <number>, <string>

An unexpected TFTP error has occurred. The file transfer has failed.

s48273: Internal error: <string>

An internal error has occurred. Contact your distributor or reseller.

s48274: At least one of SERVER, FILE, DELAY and DESTINATION required

At least one of these four parameters must be specified.

s48275: Loader module reset, active transfers aborted

The loader module has been reset. Any active transfers have been aborted.

s48276: Delay until load start is longer than two minutes

The load has been delayed by more than two minutes. This delay is specified by the DELAY=n parameter of the LOADER commands. If a shorter delay is desired, the current load should be cancelled and the command should be re-issued with a different value of the DELAY parameter. In the command, the delay is specified in seconds.

s48277: TFTP open failed

The LOADER module was unable to open a connection to the TFTP server.

s48279: Invalid file name "<string>"

The file name entered in this command was incorrect. File names must have a <name>.<typ> as the final substring, where <name> is up to 8 characters long, and <typ> is up to three characters long.

s48280: Parameter <string> not specified in default configuration or command

The specified parameter was not entered in the LOAD command, and a default value did not exist either. This means the LOAD can not proceed. Specify the missing parameter either as a default, or on the command line.

s48281: No load in progress, nothing to reset

There was no LOAD in progress, so the RESET LOAD command can not take effect.

s48285: The specified port is not valid

The specified port to load the file from is invalid, i.e. the port does not exist on this system.

s48286: The specified file type can't be uploaded

The specified file type can't be uploaded. The system can't upload *.PAT or *.REL files.

s48287: The specified file was not found on the system

The specified file was not found on the system.

s48288: File skipped <string>

The file was skipped by zmodem because it already exists on the system, or the filename is not valid.

s48289: File <string>, transferred successfully

The file was transferred successfully by zmodem.

s48290: ZMODEM, the sending host has closed the connection or is not responding

The sending host has closed the connection, or did not respond to our requests.

s48291: Upload, a file error occurred reading the file

The upload failed due to a general file error reading the file.

s48292: ZMODEM, session over

The current zmodem session is finished because one of the ends terminated it.

s48293: ZMODEM, the receiving host has closed the connection or is not responding

The receiving host has closed the connection, or didn't respond to our requests.

s48294: The specified method is not supported

The specified method for loading from a port is not supported.

s48296: HTTP open failed

The LOADER module was unable to open a connection to the HTTP server.

s48297: Load aborted: <string>

The HTTP server reported the listed error.

s48298: The specified method is not support for file uploads

The specified method is not supported for file uploads.

s49256–s49999: INSTALL**s49256: Parameter <string>, invalid file name, should be <dev>:<mod>\<fil>.<typ>**

The file name entered for the specified parameter was invalid. The correct format for file names is given. Re-enter the command with the file name in the correct format.

s49257: One or more parameters missing; required are <string>

One or more required parameters were not entered in the command. The required parameters for this command are given. Re-enter the command with the required parameters.

s49258: Invalid device for <string> file

The device specified in the command was an incorrect device, either because it is unrecognised altogether, or because it is not valid for the particular type of file being specified.

s49259: Invalid module for <string> file

The module specified in the command was incorrect, either because it is unrecognised altogether, or because it is not valid for the particular type of file being specified.

s49266: Can not specify RELEASE for DEFAULT install

The install specified was DEFAULT, but a RELEASE was specified for this. The DEFAULT release can only be EPROM, so the RELEASE parameter is not allowed for the DEFAULT install. Re-enter the command for another install (TEMPORARY or PREFERRED) or remove the RELEASE parameter.

s49268: File not found for <string>

The file specified for RELEASE was not found. Re-enter the command with the name of a file that exists, or ensure that the required file is present on the router.

s49270: Install information already deleted, no action required

On deleting an install, the information for the install was already found to be deleted. No action is required on this command.

s49271: Parameter <string>, invalid release number, should be <maj>.<min>

The parameter specified should have been a release number and was found not to be valid. Re-enter the command specifying a valid release number.

s49273: Open of specified configuration file <string> failed

The system tried to open the specified configuration file prior to writing configuration information to the file, but this open failed.

s53256–s53999: Trigger Facility**s53256: Required parameters missing, <string> is required**

One or more parameters required by the command were not supplied.

s53257: Trigger type parameter <string> must precede <string>

Any type-specific or optional parameter in a CREATE TRIGGER or SET TRIGGER command must be preceded with a trigger type parameter. You should modify the command so that the parameter immediately following TRIGGER is a trigger type parameter (TIME, REBOOT, PERIODIC, etc.).

s53258: Trigger <number> does not exist

The specified trigger does not exist.

s53259: Trigger <number> is already defined

The specified trigger is already defined. Each trigger must have a unique number.

s53260: Internal error: <string>

An unexpected internal error has occurred. Examine the router log (SHOW LOG) for further information and contact your distributor or reseller for advice or assistance.

s53261: <string> successfully deleted

The specified item has been deleted.

s53262: <string> successfully added

The specified item has been added.

s53263: Only one trigger type (TIME, PERIODIC, REBOOT, etc.) may be specified

Only one trigger type parameter (TIME, PERIODIC, REBOOT, etc.) may appear on any command line.

s53264: Trigger <number> has been activated

The specified trigger has been manually activated. Manual trigger activations do not update the activation counters, and cause actions to be performed even if the trigger is in test mode.

s53265: The trigger module is not enabled

The command can not be performed while the trigger module is disabled.

s53266: The trigger module is already enabled

The trigger module is already enabled.

s53267: The trigger module is already disabled

The trigger module is already disabled.

s53268: The trigger module has been enabled

The trigger module has been enabled. Any configured and enabled triggers will be activated as appropriate.

s53269: The trigger module has been disabled

The trigger module has been disabled. No further trigger activations will occur.

s53270: Trigger <number> is already enabled

The specified trigger is already enabled.

s53271: Trigger <number> is already disabled

The specified trigger is already disabled.

s53272: Trigger <number> configuration updated

The configuration of the specified trigger has been updated.

s53273: Maximum number of <number> scripts on trigger exceeded

The maximum number of scripts that can be associated with a single trigger has been exceeded.

s53276: Bad position number (<number>) for trigger <number>

The supplied position number is invalid.

s53277: No settings specified

No new settings were specified on the SET command.

s53278: Option <string> not available for <string> triggers

The specified option is not available for that type of trigger.

s53279: Invalid parameters: <string>

That combination of parameters is not valid.

s53280: The <string> parameter may not appear more than <number> times in a command

The specified parameter has appeared too many times in one command.

s53281: The <string> parameter may not appear multiple times in this command

The specified parameter may not appear more than once in that particular command.

s53282: Action <number> configuration updated

The configuration of the specified action has been updated.

s53283: No <string> found

No items of the requested type were found.

s53284: Action <number> is in use and can not be destroyed

The specified action is referenced (used by) at least one trigger and therefore can not be destroyed. Remove all references to this action and retry the command.

s53285: Trigger module configuration has been reset to defaults

The configuration of the trigger module has been erased and reset to defaults.

s53286: Trigger module configuration has been erased

The configuration of the trigger module has been erased and will occupy no space in nonvolatile storage.

s53287: Parameters DAYS and DATE not allowed together

If a specific date for a trigger has been set then a day of the week may not be specified. Conversely if a day or days have been specified for the activation of a trigger then a specific date may not be given.

s54256–s54999: Scripting**s54256: No script file with that name was found**

No script (SCR) file with the specified name was found. Use the command SHOW FILE to display the script files that are currently stored on the router.

s54257: An error occurred while writing the script <string> to storage

No script (SCR) file with the specified name was found. Use the command SHOW FILE to display the script files that are currently stored on the router.

s54258: The file name given for the script is invalid

The file name given for the script is invalid. The file name is not consistent with a file name of the type dev:nnnnnnnn.scp

s54259: Line number <number> has been deleted from script <string>

The specified SCR script line has been deleted.

s54260: Line number <number> can't be deleted from script <string>

The specified script line can not be deleted because it does not exist.

s54261: Script <string> has been successfully deleted

The specified script has been deleted from the system.

s54262: The parameter LINE must be specified

The parameter LINE must be specified for a SET SCRIPT command.

s54263: The parameter TEXT must be given if LINE has been specified

The command given requires a TEXT parameter to make it a valid command.

s54264: The script <string> has been deactivated

The named script has been deactivated, and stopped from further action.

s54265: The script <string> is not currently active

The named script is not currently active and hence can't be deactivated.

s55256–s55999: Time Division Multiplexing (TDM)**s55256: Parameter <string>, invalid interface <string>**

The value specified for the INTERFACE parameter should represent a valid BRI interface but doesn't. Re-enter the command with a valid BRI interface.

s55257: Parameter <string>, <string> port <number> does not exist

The specified BRI port instance is not provided by the hardware. Find out how many BRI ports the router has and re-enter the command specifying a port that exists.

s55258: Parameter <string>, the group name is too long

The GROUP parameter specified a group name that is more than 15 characters. Re-enter the command with a group name of 15 characters or less.

s55259: Parameter <string>, missing a slot number

A slot number was expected by the parsing routine but another character was found instead. Check the format of the slot list and re-enter the command with a valid slot list.

s55260: Parameter <string>, contains an invalid slot number

A slot number of more than 31 has been specified. Re-enter the command and specify only slot numbers between 1 and 31.

s55261: Parameter <string>, contains an invalid character

An invalid character has been found by the slot list parsing routine. Re-enter the command with a valid slot list.

s55262: Parameter <string>, contains an invalid slot range

A slot range with invalid bounds has been specified. Re-enter the command and specify valid bounds for each slot range.

s55263: Parameter <string>, a slot number is specified twice

A slot number has been specified in more than one slot list entry. Re-enter the command and make sure each slot is only specified once.

s55264: Parameter <string> is required

The specified parameter is required by this command but was not included. Re-enter the command specifying the missing parameter.

s55265: No TDM groups are defined

No TDM groups have been defined, so no action can be taken.

s55268: Group <string> does not exist

The group name specified by the GROUP parameter does not exist. Use the CREATE TDM GROUP command to create the group.

s55269: Group <string> already exists

A group with the same name already exists. Re-enter the command with a different group name.

s55270: <string> port <number> is set to ISDN mode

The specified BRI port has not been set to TDM mode or MIXED mode. Use the SET BRI commands to set the mode.

s55271: Some or all specified slots are not used by group <string>

The specified group does not use one or more of the slots specified by the SLOTS parameter.

s55272: No TDM groups are defined for <string> port <number>

No TDM groups have been defined for the specified BRI port, so no action can be taken.

s55273: Group <string> is not defined for <string> port <number>

The specified group is not defined over the specified BRI port. It may be defined for another BRI port.

s55274: A user is attached to group <string>

An attempt was made to destroy a TDM group that has a user module attached to it. Destroy the user module instance that is attached to the TDM group and re-enter the command.

s55275: Can not delete all slots when a user is attached

An attempt was made to destroy a TDM group that has a user module attached to it. Destroy the user module instance that is attached to the TDM group and re-enter the command.

s55276: Can not add any more groups to this interface

An attempt was made to add a new group to an interface which has already reached the maximum number of groups possible.

s55278: BRI interfaces do not have an unstructured mode

An attempt has been made to put a BRI interface into unstructured mode. Unstructured mode is only possible on a PRI interface.

s55280: The parameters SLOTS and UNSTRUCTURED may not be specified together

The parameters SLOTS and UNSTRUCTURED are mutually exclusive, they may not be specified together. Slots have no meaning when an interface is in unstructured mode.

s56256–s56999: File Subsystem**s56256: No viewer is currently assigned to this file type**

There is no viewer currently assigned to this file type.

s56257: The specified file was not found on the system

The specified file could not be found.

s56258: The specified file could not be opened

The specified file could not be opened. This is because of a storage subsystem failure.

s56259: The specified file is not a text file

The specified file type indicates that the file is not a text file. This would be true of release files.

s56260: Source and destination filenames were not found

The file command expected a source and destination filenames.

s56261: The source and destination filenames must be on the same device

The rename command requires the source and destination filenames to be on the same storage device.

s56262: The source and destination filenames must have the same file type extension

The rename command requires the source and destination filenames to have the same filename type extension.

s56263: A file with the same destination filename already exists

The rename command requires the destination filename to be unique.

s56264: The filename given is invalid

The given filename is not valid, for some reason, such as being too long or having a bad file type.

s56265: file <string> not created, insufficient device space or write error

The file creation failed either because there was insufficient device space or a write of the file header to the device failed.

s56266: file <string> not completely written due to write error

The write of data to a file failed due to a hardware failure.

s56267: an error occurred opening file <string>

An unspecified error occurred opening the file.

s56268: an error occurred reading file <string>

An unspecified error occurred while reading the file.

s56269: an error occurred while trying to delete the file <string>

An unspecified error occurred while trying to delete the file.

s56270: an error occurred while trying to rename the file <string>

An unspecified error occurred while trying to rename the file.

s56271: the contents of file <string> are invalid

While reading the specified file, it was found the contents are invalid.

s57256–s57999: Logging Facility

s57256: Internal Error: <string>

An unexpected internal error has occurred. Contact your distributor or reseller.

s57257: <string> is already enabled

The specified feature or function is already enabled.

s57258: <string> has been enabled

The specified feature or function has been enabled.

s57259: <string> is already disabled

The specified feature or function has already been disabled.

s57260: <string> has been disabled

The specified feature or function has been disabled.

s57261: Specified output definition does not exist

The output definition referenced by the command does not exist. Use the SHOW LOG OUTPUT command to see a list of output definitions.

s57262: Required parameter(s) missing, <string> required

One or more parameters required by the command were not supplied. Please reissue the command with all required parameters, and consult the online help (HELP LOG), online or printed reference manuals for further assistance.

s57263: Output definition is already defined

You attempted to create an output definition that already exists. If you wish to modify the settings of the existing output definition, use SET LOG OUTPUT. If you wish to create a new output definition, you must select an index number for it that is not currently in use. Use SHOW LOG OUTPUT to see a list of existing output definitions.

s57264: DESTINATION=<string> only permitted when OUTPUT=<string>

The destination value in your command can only be used on certain output definitions.

s57265: <string> successfully created

The specified object has been created successfully.

s57266: <string> successfully destroyed

The specified object has been successfully destroyed.

s57267: No changes specified

Your SET command did not specify any new parameter values.

s57268: Filter <number> does not exist on specified output definition

The specified filter does not exist.

s57269: Netmask <ipadd> is incompatible with address/network <ipadd>

You have specified a network mask that is not compatible with the given IP address or network.

s57270: Cannot specify MASK without ORIGIN

The MASK parameter may not be used unless the ORIGIN parameter also appears in the same command.

s57271: ALL cannot be specified with other conditions

You can not mix ALL with other conditions.

s57272: New filter number for output definition <number> cannot exceed <number>

The new filter number specified by your command is not acceptable. The largest acceptable filter number is quoted by the error message.

s57273: <string> added successfully

The specified object has been added to the log module configuration.

s57274: <string> deleted successfully

The specified object has been deleted from the log module configuration.

s57275: Output definition has no filters attached

The specified output definition does not have any filters configured.

s57276: The log module has been reset

The log module has been reset.

s57277: The log module is not enabled

The log module is not enabled. This command can only be issued when the log module is enabled. To enable the log module, use ENABLE LOG.

s57278: The log module configuration has been reset to defaults

The log module configuration has been erased and reset to default values. If the PURGE LOG command is issued when the log module is disabled, the configuration is erased instead of being reset to default values.

s57280: FULL can not be mixed with FILTER

The FULL parameter can not be specified with the FILTER parameter.

s57281: FILTER can only be used with a specific output definition

The FILTER parameter may only be used when a specific output definition is referenced, by providing a value on the OUTPUT parameter.

s57282: The "<value>" search modifier can only be used with string values

The "%" search modifier searches for one string within another, so it can only be used with string values.

s57283: Illegal date: <string>

The date value supplied in the command is incorrectly formatted. Valid forms are DD-MMM-YY (e.g. 29-FEB-1996) and DD/MM/YY (e.g. 29/03/96).

s57284: Illegal time: <string>

The time value supplied in the command is incorrectly formatted. The correct form is HH:MM:SS (eg. 15:04:00) with both the minute and second parts being optional. (eg. 13, 13:00 and 13:00:00 are all legal and refer to 1pm).

s57285: Specified log reception entry already exists

The command attempted to create a log reception entry that already exists.

s57286: Cannot specify MASK when RECEIVE=ANY

The MASK parameter can not be used when RECEIVE=ANY.

s57287: Log reception entry does not exist

The specified log reception entry does not exist.

s57288: Cannot specify PASSWORD when PROTOCOL=OLD

A password can not be specified for the old router log message protocol, since that protocol does not support password protection.

s57289: SUBTYPE can only appear after TYPE has been specified

The TYPE parameter must always appear in a command before the SUBTYPE parameter may be specified.

s57290: Parameter(s) invalid for DESTINATION=<string>

Some parameter(s) in the command can not be used with the specified value of DESTINATION.

s57291: Syntax error - nothing may follow "=*"

The PARAMETER=* syntax is used to remove a condition from a log message filter. This means that the parameter may take any value. It does not make sense to write anything after the "*".

s57292: No (matching) log messages found

No matching log messages were found.

s57293: Can not have DESTINATION=<string> when OUTPUT=<string>

The specified value of DESTINATION can not be used with that output definition.

s57294: Port <number> is already in use

The specified port is already being used.

s57295: The IP module is disabled, can not send/receive ROUTER/SYSLOG messages

The IP module is disabled, therefore the log system is unable to send or receive any IP traffic such as SYSLOG and router-to-router (SRLP, Net Manage) messages. If these functions are desired, the IP module should be enabled and correctly configured.

s57296: The UTC offset has been set

The offset between this routers local time and UTC/GMT has been changed.

s57297: ACTION must be specified with other conditions

A log filter can not be added with an ACTION only condition.

s57298: A match all filter has fixed ACTION=PROCESS

An ALL filter can not have its action changed to IGNORE.

s58256–s58999: PING**s58256: <string> is currently active**

This message is displayed if user is trying to make another ping or trace session while there is an active session already running.

s58257: <string> module is not active

This message is displayed if user is trying to execute a particular type of PING, but the network layer for the session, i.e. IP is not enabled.

s58258: Illegal parameter combination, refer to online HELP (HELP PING)

The command supplied by the user contains a syntax error.

s58259: <string>, finished successfully

The ping or trace session finished successfully.

s58260: The destination is either un-specified or invalid

The ping or trace session does not contain a valid IP address.

s58261: Invalid source: <ipadd>, default source address <ipadd> will be used

This message is displayed as a warning to a trace user that there is no source address to be used, and the PING module automatically chooses the address of one of the IP interfaces configured as the source address of the trace packets.

s58262: No more than <number> packets may be sent continuously

This message is displayed if user is trying to send more than a certain number of packets in continuous fashion. This limitation is placed to avoid the PING module monopolising the CPU time in such a way that it degrades the performance of other modules.

s58263: Specified host is not in the host table

This message is displayed when the user specifies a host name as a replacement for an IP address, but the host name is not listed in the host table. See the ADD IP HOST command for more information about adding a host name to the host table.

s58264: No <string> info for this device

This message is displayed when the user is trying to display the PING session information, but there is currently no PING information available.

s58265: Other logged-in users still using the PING and/or TRACE

This message is displayed if a privileged user is trying to set or purge the default PING/TRACE parameters, while other logged-in users are still using their sessions.

s58266: The current <string> information has been purged by device <number> (<string>)

This message is displayed on all users logged-in and using the PING/TRACE service while a privileged user is forcing the PING module to purge all default information regardless.

s58268: Invalid IP source address <ipadd>

This message is displayed if the specified IP source address is not one of our IP interfaces' address.

s58270: No route to destination <ipadd>

The ping could not be transmitted because no route exists to the specified destination.

s58271: Target/host <ipadd> is unreachable

The target of the ping is unreachable. No response has been received from the target.

s58273: Incompatible source and destination types

The specified source and destination types are incompatible. Re-enter the command with a valid source and destination.

s58274: The timeout must be less than or equal to 60 seconds

An invalid timeout was specified. Re-enter the command with a valid timeout.

s59256–s59999: Simple Network Management Protocol (SNMP)**s59256: SNMP community creation failed**

The creation of an SNMP community failed due to some reason. Check all parameters entered on the command line and re-enter the command.

s59257: SNMP community <string> not found

The given SNMP community was not found, so the command could not be executed. Re-enter the command with the name of an existing SNMP community.

s59258: No action specified on command line

A command was entered which actually causes no action. No error has occurred, but no action has taken place either. Re-enter the command but specify an action.

s59259: SNMP community addition failed

Adding a trap host or manager to an SNMP community failed. The usual reason for this is that the trap host or manager is already defined for that community.

s59260: SNMP community deletion failed

Deleting a trap host or manager from an SNMP community failed. The usual reason for this is that the trap host or manager was not defined for that community.

s59261: SNMP community destruction failed

Destroying an SNMP community failed. The usual reason for this is that the community does not exist.

s61256–s61999: Telephony Services**s61256: No group with that name was found**

No group with the specified name was found. Use the command `SHOW PBX GROUP` without a group name specified to display the groups that are currently defined.

s61257: One or more of the extensions in the group is ringing

The group is in alert mode, as one or more extensions in the group is ringing.

s61258: The parameter EXTENSION is required to create a group

The parameter `EXTENSION` is a required parameter for the `CREATE GROUP` command.

s61259: The provided extension, or part of the extension, is already in use

The extension is checked to ensure it is unique. The check makes sure that both the extension and part of the extension are unique.

s61260: A group with the given name already exists

The group with the given name already exists.

s61261: An illegal combination of parameters has been given

An illegal combination of parameters has been given. Refer to the Reference Manual for the correct syntax.

s61262: An expected parameter was not found in the command

A required parameter was not found on the command line.

s61263: The given prefix or extension is invalid

The given prefix or extension is invalid.

s61264: The given number is invalid or illegal in its current usage

The given number is invalid.

s61265: The given extension does not exist

The given extension does not exist and hence is invalid.

s61266: The given shortcode does not exist

The given shortcode does not exist and hence is invalid.

s61267: The specified port is already allocated to another extension

The specified port is already allocated to another extension, and hence is not free to be used again.

s61268: The specified port is off-hook or is invalid

The specified port is either off-hook or invalid, and hence can't be added to an extension.

s61269: The given extension number to copy does not exist

The given extension number to copy does not exist, and hence the copy will fail.

s61270: A shortcode with the given index already exists

A shortcode with the given index already exists, and hence the command has failed.

s61271: A default extension can't be destroyed

The user can't destroy a default extension.

s61272: The specified port does not exist

The user specified a port greater than the number of voice ports on the router.

s61273: The given extension number already exists

The given extension number already exists, and hence the command has failed.

s61274: <string> is not enabled

The specified feature is not enabled, so the operation can not be carried out. Some features in the router require that the feature be enabled before they can be configured. Re-enter the command after enabling the feature.

s61275: <string> is already enabled

The specified feature is already enabled, so the operation does not need to be carried out.

s61276: <string> is already disabled

The specified feature is already enabled, so the operation does not need to be carried out.

s61277: This router has no VOX ports

The router does not have any VOX ports so all PBX commands have been disabled.

s70256–s70999: Dynamic Host Configuration Protocol (DHCP)

s70256: There are no free UDP ports available for DHCP

DHCP has tried to open a UDP listen port for DHCP but failed due to a lack of free UDP listen ports

s70257: The given range of IP addresses is too large

The given range is too large. Choose a smaller range of IP addresses.

s70258: The given range of IP addresses overlap with an existing range

The given range of IP addresses overlap with an existing range. Redefine the range so it does not overlap.

s70259: The given IP address is not in the given range

The given IP address is not in the IP address range of the specified range.

s70260: policy <string> is still inuse by host <ipadd>

The given policy can't be destroyed as a host is still using it.

s70261: <string> is already enabled

An attempt was made to enable the specified object, but it is already enabled. The command is redundant.

s70262: <string> is already disabled

An attempt was made to disable the specified object, but it is already disabled. The command is redundant.

s70263: unable to write range database file

An attempt to write the range database file failed.

s70264: static entry with same hardware address exists

An attempt was made to add the same static DHCP client twice.

s77256–s77999: Firewall**s77256: Interface <value> was not found in the security policy <string>**

The specified interface was not found in the list of interfaces currently known to the specified security policy.

s77257: 070707<string>

Sends a message with a bell to the manager terminals.

s77258: The lower value of the range is greater than the top value

The values given for the range are invalid. The lower value for the range is greater than the top value for the range.

s77259: A public interface can only be present in up to two policies

An attempt to add a public interface has failed. A public interface can appear in a maximum of two different policies.

s77260: The specified filename is invalid or not of type text

The given filename is not valid. Either the file is not in the correct format or the file type is not text.

s77261: File <string> does not exist

The given file does not exist on the router.

s77262: Port number given is not valid

A port number must always be specified when adding a port notify. The port specified must be present on the router. Use the command SHOW PORT SUMMARY to display the ports available.

s77263: Protocol given is not valid in this situation

The protocol value given is not applicable in this situation. This could be because it is illegal to add a filter with a protocol of ICMP.

s77264: Too many entries in table

The user has tried to add too many entries to the table.

s77265: A feature license is required to enable the FIREWALL module

A feature license is required before the firewall can be enabled.

s77266: Interface <value> is not set to method application in policy <string>

The specified interface is not set to method APPLICATION in the specified security policy.

s77267: A feature license is required to add FIREWALL proxies

A feature license is required before firewall proxies can be added.

Appendix B

Reference Tables

Module Identifiers and Names	B-2
FLASH File System Message Codes	B-4
ISDN Q.931 Call Clearance Cause Codes	B-6
Log Message Types and Subtypes	B-8

Module Identifiers and Names

Table B-1 on page B-2 lists the module identification number (ID), predefined name and description for all software modules in the router. The predefined name can be used instead of the module ID to identify modules in commands that require a module ID, such as the log message filter commands in the Logging Facility (see *Chapter 12, Logging Facility*).

Table B-1: Module Identifiers, Names and Descriptions.

ID	Name	Description
0	NONE	None.
3	PPP	Point-to-Point Protocol (PPP), the Internet standard for router-to-router connections over point-to-point data links.
5	IP, IPG	Internet Protocol (IP) routing, including RIP and EGP routing protocols, ICMP, UDP and SNMP.
14	Q931	Q.931 ITU-T standard for ISDN call control.
18	TEST	Hardware testing facility built in to the router for testing all network interfaces and coprocessors on the router.
19	LAPD	LAPD data link layer for ISDN D channel access.
22	TCP	Transmission Control Protocol (TCP), the Internet standard transport protocol.
23	ETH	Ethernet device drivers and logical link layer.
25	TS, TSERVER	Terminal server.
28	COMP	Compression module, the code that runs on the router's CPU to control the compression.
30	X25T	X.25 DTE (layer 3) handler, provides network communications over an X.25 network.
31	FLASH	FLASH device drivers.
33	TLNT, TELNET	Telnet protocol for terminal connections over IP into and out of the router.
34	SYS, SYSTEM	General system functions.
35	CH	Command processor.
36	TTY	Terminal drivers used by Telnet and asynchronous ports.
37	ICC, ISDNCC	ISDN call control, responsible for maintaining and controlling ISDN calls.
38	MIOX	Multiprotocol Interconnect Over X.25 (MIOX), enables network layer protocols to use X.25 in a standard way.
39	BOOT	BOOTP boot protocol, assists devices attempting to make BOOTP requests over a wide area network.
41	BRI	ISDN Basic Rate Interface device drivers.
43	PORT	Asynchronous port device independent layer.
45	USER	User login handler, including the User Authentication Database and TACACS support.
47	ASYN	Asynchronous device independent drivers.
48	LOAD	Software release, patch and script loading.
49	INST, INSTALL	Handles installation of software at boot from ROM, FLASH and NVS.

Table B-1: Module Identifiers, Names and Descriptions. (Continued)

ID	Name	Description
53	TRG, TRIGGER	Trigger facility for executing router commands in response to events.
54	SCR	Scripting module.
55	TDM	Time Division Multiplexing (TDM) module for running serial links over a Primary Rate Interface.
56	FILE	File subsystem that uses both BBR and FLASH. Used mainly for scripts.
57	LOG	Logging facility that handles all router logging messages.
58	PING	Universal ping function.
59	SNMP	SNMP agent.
60	SCC	SCC drivers.
61	PBX	Telephony services (POTS).
69	HOSTMIB	Host Resources MIB.
70	DHCP	Dynamic Host Configuration Protocol.
71	INT	Interfaces module.
73	ENCO	Compression services.
77	FIREWALL	Nemesis firewall.

FLASH File System Message Codes

Some FFS processes generate messages in the system log (displayed with the SHOW LOG command on page 12-31 of *Chapter 12, Logging Facility*) which include FLASH File System (FFS) message codes. Table B-2 on page B-4 lists the possible codes and their meanings.

Table B-2: FLASH File System Message Codes.

Code	Meaning
0	Operation successful.
1	Writing suspended until compaction completes.
2	Writing suspended until erasure completes.
3	Writing suspended until verification completes.
4	Reading suspended until compaction completes.
5	Reading suspended until erasure completes.
6	Reading suspended until verification completes.
7	Creation suspended until compaction completes.
8	Creation suspended until erasure completes.
9	Creation suspended until verification completes.
10	Deletion suspended until compaction completes.
11	Deletion suspended until erasure completes.
12	Deletion suspended until verification completes.
13	Insufficient memory space to create file.
14	File already exists - cannot create new file with same name.
15	File deleted - cannot write to a deleted file.
16	File closed - cannot write to a closed file.
17	File opened to read - cannot write.
18	Cannot write past end of file.
19	File already closed - cannot close.
20	File does not exist - cannot open.
21	Invalid file header check sum.
22	Invalid file data check sum.
23	File deleted - cannot read deleted file.
24	File closed - cannot read closed file.
25	File opened for write - cannot read.
26	Cannot read past end of file.
27	File does not exist - cannot delete.
28	File not found.
29	Device erasure failed.
30	Compaction suspended until erasure complete.
31	Verify suspended until compaction complete.
32	Verify suspended until erasure complete.
33	Erase already in progress.
34	Compaction already in progress.

Table B-2: FLASH File System Message Codes. (Continued)

Code	Meaning
35	Verification already in progress.
36	Out of memory.
37	Hardware failure has occurred.
38	Invalid operation to complete file open to read.
39	Invalid operation to close file open to write.
40	Getting file to undersized buffer.
41	File already deleted - cannot delete.
42	File not complete - cannot read incomplete file.
43	File data corruption detected.
44	Restart underway - all access is suspended.
45	FLASH device is currently busy with a previous command.
46	File does not exist - cannot check.
47	File not complete - cannot check.
48	File check sum invalid - cannot open.
49	File too large for get - use open and read.
50	No FLASH is present.
51	Operation failed - no programming power available.

ISDN Q.931 Call Clearance Cause Codes

Table B-3 on page B-6 lists ISDN Q.931 call clearance cause codes for Q.931 call control profiles currently supported by the router. Not all cause codes are supported by all ISDN service providers.

Table B-3: ISDN Q.931 Call Clearance Cause Codes and Descriptions.

Code	Description
1	Unallocated
2	No route to specified transit network
3	No route to destination
6	Channel unacceptable
7	Call awarded and being delivered in an established channel
16	Normal call clearing
17	User busy
18	No user responding
19	No answer from user (user alerted)
21	Call rejected
22	Number changed
26	Non-selected user clearing
27	Destination out of order
28	Invalid format number
29	Facility rejected
30	Response to STATUS ENQUIRY
31	Normal, unspecified
34	No circuit/channel available
38	Network out of order
41	Temporary failure
42	Switching equipment congestion
43	Access information discarded
44	Requested circuit/channel not available
47	Resources unavailable, unspecified
49	Quality of service unavailable
50	Requested facility not subscribed
53	Outgoing calls barred within CUG
55	Incoming calls barred within CUG at destination
57	Bearer capability not authorised
58	Bearer capability not presently available
62	Inconsistency in designated outgoing access information and subscriber class
63	Service or option not available, unspecified
65	Bearer capability not implemented
66	Channel type not implemented
69	Requested facility not implemented

Table B-3: ISDN Q.931 Call Clearance Cause Codes and Descriptions. (Continued)

Code	Description
70	Only restricted digital information bearer capability is available
79	Service or option not implemented, unspecified
81	Invalid call reference value
82	Identified channel does not exist
83	A suspended call exists but this call identity does not
84	Call identity in use
85	No call suspended
86	Call having the requested call identity has been cleared
87	Destination not member of CUG
88	Incompatible destination
90	Non-existent CUG
91	Invalid transit network selection
95	Invalid message, unspecified
96	Mandatory information element is missing
97	Message type non-existent or not implemented
98	Message not compatible with call state or message type non-existent or not implemented
99	Information element non-existent or not implemented
100	Invalid information element contents
101	Message not compatible with call state
102	Recovery on timer expiry
111	Protocol error, unspecified
127	Interworking, unspecified

Log Message Types and Subtypes

Table B-4 on page B-8 lists the log message types and subtypes defined by the Logging Facility. Commands in the Logging Facility will recognise either the numeric identifier (ID) or the textual name. See *Chapter 12, Logging Facility* for more detailed information about these identifiers.

Table B-4: Log Message Types and Subtypes.

Type ID/Name	Subtype ID/Name	Meaning
000/NULL		No type or subtype information.
	000/NULL	Used for messages issued through the old logging system which does not support log message types. Displayed as a blank in output.
001/REST		Router reboot.
	001/NORM	Ordinary restart (default case).
	002/CRASH	Router restarted after a crash.
	003/FAIL	Restart or self-test failure.
002/PINT		State change or error condition on a physical-layer interface (e.g. bri0, port3, eth0).
	001/UP	Physical interface is now available.
	002/DOWN	Physical interface is no longer available.
	003/WARN	Possible problem with physical interface.
	004/ERROR	Problem detected with physical interface.
	005/RESET	Physical interface has been reset.
003/CALL		State change or error condition on an ISDN call.
	001/UP	Call has been established.
	002/DOWN	Call has been cleared.
	003/WARN	Possible problem with call.
	004/ERROR	Problem detected with call.
	005/RESET	Call has been reset.
004/DLINK		State change or error condition on a data-link layer entity (e.g. LAPD).
	001/UP	Data-link layer entity is available.
	002/DOWN	Data-link layer entity is not available.
	003/WARN	Possible problem with data-link layer entity.
	004/ERROR	Problem detected with data-link layer entity.
	005/RESET	Data-link layer entity has been reset.
	008/ACT	Data link activated.
	009/DEACT	Data link deactivated.
005/VINT		State change or error condition on a virtual interface (e.g. ppp0).
	001/UP	Virtual interface is operational.
	002/DOWN	Virtual interface is not operational.
	003/WARN	Possible problem with virtual interface.
	004/ERROR	Problem detected with virtual interface.
	005/RESET	Virtual interface has been reset.
	006/ACT	On-demand interface has been activated.
	007/CREATE	Virtual interface has been created.
	008/DEST	Virtual interface has been destroyed.

Table B-4: Log Message Types and Subtypes. (Continued)

Type ID/Name	Subtype ID/Name	Meaning
006/CIRC		State change or error condition on a circuit (e.g. X.25 circuit, PPP control protocol).
	001/UP	Circuit is operational.
	002/DOWN	Circuit is not operational.
	003/WARN	Possible problem with circuit.
	004/ERROR	Problem detected with circuit.
	005/RESET	Circuit has been reset.
	006/CONF	Automatic configuration/option negotiation.
007/ATT		State change or error condition on an attachment (module interface).
	001/ATTCH	Module has attached.
	002/DETC	Module has detached.
	003/FAIL	Module attachment failed.
008/EXCEP		Unexpected exception detected.
	000/RESET	Router reset.
	001/EXTNO	External contact open.
	002/EXTNC	External contact closed.
	003/TNORM	Temperature normal.
	004/THIGH	Temperature threshold exceeded.
	005/TREND	Temperature trend exceeded.
	008/BUS	Bus error.
	012/ADDR	Address error
	016/INSTR	Illegal instruction
	032/PRIV	Privilege violation
	040/LINEA	Line A emulator
	044/LINEF	Line F emulator
	096/SPUR	Spurious interrupt
	128/TRAP0	Trap #0 (fatal)
	132/TRAP1	Trap #1 (restart)
	136/TRAP2	Trap #2 (assert)
009/BUFF		Significant change in the number of free buffers.
	001/LEV1	Number of free buffers dropped below buffer level 1.
	002/LEV2	Number of free buffers dropped below buffer level 2.
	003/LEV3	Number of free buffers dropped below buffer level 3.
011/AUTH		Authentication requests or security issues.
	001/OK	Successful authentication (login, connect, etc.).
	002/FAIL	Unsuccessful authentication.
	003/RFAIL	Repeated unsuccessful authentication.
012/BATCH		Trigger Facility/Scripting activity.
	001/ACT	Trigger/Script activation.
	002/CMD	Trigger/Script command.
	003/OUT	Trigger/Script output.

Table B-4: Log Message Types and Subtypes. (Continued)

Type ID/Name	Subtype ID/Name	Meaning
015/SYSLOG		Messages received via syslog.
	000/KERN	Kernel messages.
	008/USER	Various user-level messages.
	016/MAIL	Mail subsystem.
	024/DAEMON	System daemons.
	032/AUTH	Security/authorisation messages.
	040/SYSLOG	Messages generated internally by syslogd.
	048/LPR	Line printer subsystem.
	056/NEWS	Network news subsystem.
	064/UUCP	UUCP subsystem.
	072/CRON	Clock daemon.
	080/AUTHPRIV	Private security/authorisation messages.
	128/LOCAL0	<i>Reserved for local use.</i>
	136/LOCAL1	<i>Reserved for local use.</i>
	144/LOCAL2	<i>Reserved for local use.</i>
	152/LOCAL3	<i>Reserved for local use.</i>
	160/LOCAL4	<i>Reserved for local use.</i>
	168/LOCAL5	<i>Reserved for local use.</i>
	176/LOCAL6	<i>Reserved for local use.</i>
	184/LOCAL7	<i>Reserved for local use.</i>
018/FLASH		Problems with FLASH memory.
019/USER		User activity.
	001/LON	User logon.
	002/OFF	User logoff.
	003/ADD	User account added.
	004/DEL	User account deleted.
	005/PWCHG	User password changed.
	006/PWERR	Error setting manager password.
	007/PWSET	Manager password successfully set.
	008/LOOP	Loopback problem at login prompt.
	009/TACQ	TACACS request.
	010/TACR	TACACS response.
	011/LFAIL	User logon failure.
020/CMD		Command processing.
	001/MGR	Manager command.
	002/USER	User command.
021/MSG		Router messages.
	001/INFO	Informative (Info) message.
	002/WARN	Warning message.
	003/ERROR	Error message.
022/CONFIG		Router or network configuration information/warnings.
	001/TOPO	Network topology issue.
	002/NTNUM	Conflicting network numbering (e.g. IPX, AppleTalk).
	003/NTNAM	Conflicting network name (e.g. AppleTalk).

Table B-4: Log Message Types and Subtypes. (Continued)

Type ID/Name	Subtype ID/Name	Meaning
023/IPFILT		IP filter matches.
	001/PASS	IP packet passes filter.
	002/FAIL	IP packet fails filter.
	003/DUMP	IP packet dump.
	004/FRAG	IP packet discarded by fragment filtering.
	005/SA	IP packet discarded by security association.
	006/SRCRT	IP source route packet discarded.
	007/RECRT	IP <i>record route</i> packet forwarded.
024/INTERR		Unexpected internal error.
	001/BDPKT	Bad packet detected within system code.
	002/IVPAR	Invalid parameters detected.
	003/BDATT	Attach to lower layer failed.
026/LIMIT		Internal limit exceeded.
	001/IPXSV	IPX service table full.
	002/IPXRT	IPX route table full.
	003/SWCMP	All available software compression channels in use.
027/DHCP		Dynamic Host Configuration Protocol.
	001/BIND	IP address allocated to device.
	002/FREE	Device relinquished IP address.
	003/FAIL	Refused to allocate IP address to device.
028/PBX		PBX (Telephony Services).
	001/OIF	Outbound call, internal failure.
	002/ONF	Outbound call, network failure.
	003/OOK	Outbound call, completed OK.
	004/IIF	Inbound call, internal failure.
	005/INF	Inbound call, network failure.
	006/IOK	Inbound call, completed OK.
	007/OVER	Override
	008/POVER	High priority override
	009/HOOK	Extension on/off hook
	010/FEAT	PBX feature enabled.
031/ENCO		ENCO.
	002/STACSW	STAC SW subsystem.

Table B-4: Log Message Types and Subtypes. (Continued)

Type ID/Name	Subtype ID/Name	Meaning
036/FIREWALL		Firewall Gateway.
	001/INATCP	Inbound TCP connection initiated.
	002/INAUDP	Inbound UDP connection initiated.
	003/INAICMP	Inbound ICMP connection initiated.
	004/INAOTHER	Inbound Other IP connection initiated.
	005/OUTATCP	Outbound TCP connection initiated.
	006/OUTAUDP	Outbound UDP connection initiated.
	007/OUTAICMP	Outbound ICMP connection initiated.
	008/OUTAOTHER	Outbound Other IP connection initiated.
	009/INDTCP	Denied inbound TCP connection.
	010/INDUDP	Denied inbound UDP connection.
	011/INDICMP	Denied inbound ICMP connection.
	012/INDOTHER	Denied inbound Other IP connection.
	013/OUTDTCP	Denied outbound TCP connection.
	014/OUTDUDP	Denied outbound UDP connection.
	015/OUTDICMP	Denied outbound ICMP connection.
	016/OUTDOTHER	Denied outbound Other IP connection.
	017/ATTACK	Firewall under attack.
	018/ENABLE	Firewall has been enabled.
	019/DISABLE	Firewall has been disabled.
	020/DESTROY	A firewall policy has been destroyed.
044/BOOTP		BOOTP auto-discovery feature
	001/ETHCONF	Configuration of the Ethernet interface.
045/HTTP		HTTP server
	001/GETOK	Successful GET request.
	002/GETFAIL	Failed GET request.
	003/CONFIG	Configuration attempt.

Appendix C

SNMP MIBs

Introduction	C-2
Allied Telesyn Enterprise MIB	C-3
The Products Sub-tree	C-3
The AT Router Sub-tree	C-4
The Objects Group	C-4
The arInterfaces Group	C-6
The Modules Group	C-7
MIB-II MIB	C-11
Implementation	C-12
Ethernet-like Interface Types MIB	C-13
Implementation	C-13
Host Resources MIB	C-14
Implementation	C-15

Introduction

This appendix describes the *Management Information Bases (MIBs)* and managed objects implemented by the AR Router, including any variations from the RFC standards. The router supports the following MIBs:

Table C-1: MIBs supported by the AR Router.

SNMP MIB	Description
AR Router Enterprise MIB	Objects in the Allied Telesyn Enterprise MIB for managing the AR Router.
MIB-II	The core set of objects for TCP/IP internets.
Ethernet-like Interface Types MIB	Objects for managing Ethernet-like interfaces.
Host Resources MIB	Objects for managing host systems.

Allied Telesyn Enterprise MIB

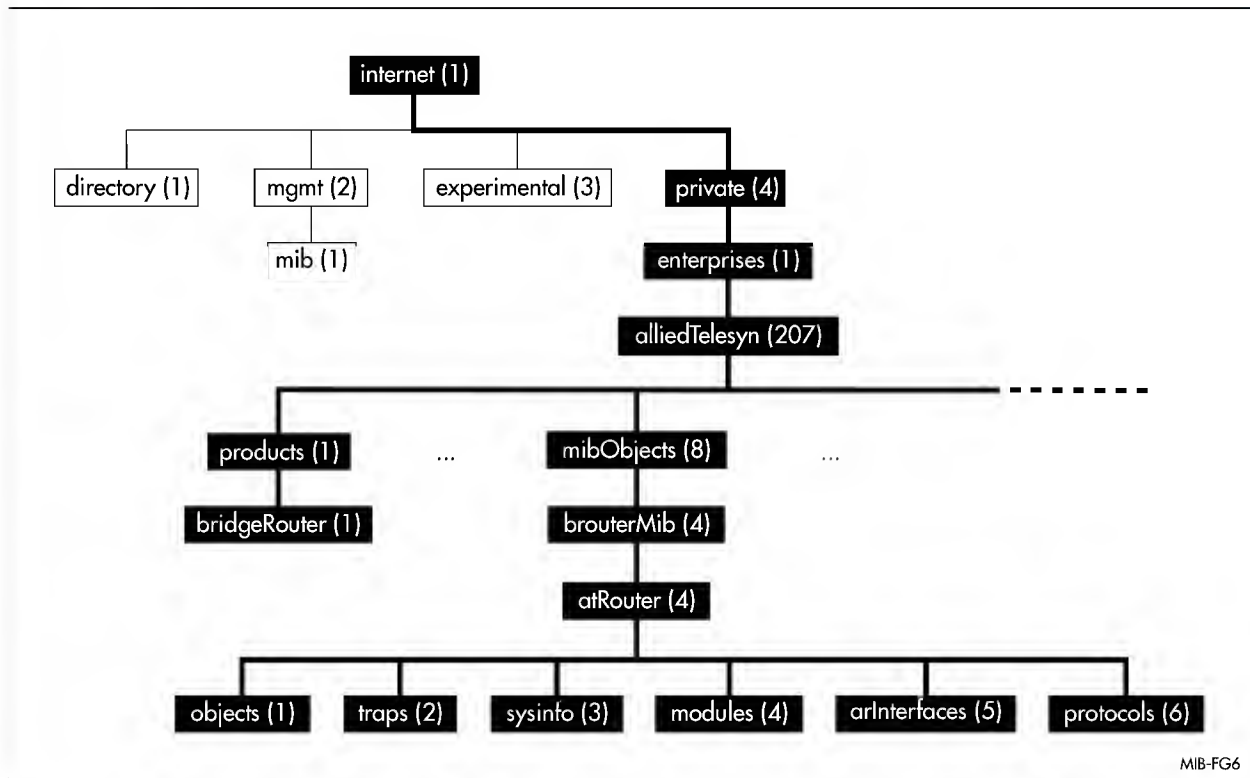
The *Allied Telesyn Enterprise MIB* defines a portion of the Management Information Base (MIB) for use with network management protocols in TCP/IP-based internets. In particular, it defines objects for managing Allied Telesyn products. Only that portion of the Allied Telesyn Enterprise MIB relating to the AR Router is described here. The AR Router supports a number of standard MIBs. Objects in the Enterprise MIB represent features specific to the AR Router that are not covered by other standard MIBs.



Not all objects in the Enterprise MIB are supported by all AR router models. For example, objects that relate to physical interfaces are only supported by AR router models that support the particular physical interface type.

Objects defined in this MIB reside in the private(4) sub-tree (Figure C-1 on page C-3) and have the object identifier prefix *alliedTelesyn* ({enterprises 207}).

Figure C-1: The Allied Telesyn Enterprise MIB sub-tree of the Internet-standard Management Information Base (MIB).



The Products Sub-tree

The Products sub-tree contains a set of object identifiers for Allied Telesyn products. Objects have the identifier prefix of *products* ({alliedTelesyn 1}). Within this sub-tree, objects with the identifier prefix *bridgeRouter* ({products 1}) are specific to the AR Router family (Table C-2 on page C-4).

Table C-2: Object identifiers for Allied Telesyn AR router products.

Object	Object Identifier	Description
bridgeRouter	{ products 1 }	
centreCOM-AR300Router	{ bridgeRouter 8 }	AT-AR300 Basic Rate ISDN (S/T interface) Router with 2 voice ports
centreCOM-AR720Router	{ bridgeRouter 11 }	AT-AR720 Modular Network Access Platform
centreCOM-AR300LRouter	{ bridgeRouter 12 }	AT-AR300L Basic Rate ISDN (S/T interface) Router
centreCOM-AR310Router	{ bridgeRouter 13 }	AT-AR310 Basic Rate ISDN (S/T interface) Router with 4 voice ports
centreCOM-AR300LURouter	{ bridgeRouter 14 }	AT-AR300LU Basic Rate ISDN (U interface) Router
centreCOM-AR300URouter	{ bridgeRouter 15 }	AT-AR300U Basic Rate ISDN (U interface) Router with 2 voice ports
centreCOM-AR310URouter	{ bridgeRouter 16 }	AT-AR310U Basic Rate ISDN (U interface) Router with 4 voice ports
centreCOM-AR350Router	{ bridgeRouter 17 }	AT-AR350 Synchronous Router
centreCOM-AR370Router	{ bridgeRouter 18 }	AT-AR370 Synchronous/Basic Rate ISDN (S/T interface) Router
centreCOM-AR330Router	{ bridgeRouter 19 }	AT-AR330 Synchronous/Dual Ethernet Router
centreCOM-AR395Router	{ bridgeRouter 20 }	AT-AR395 E1/Primary Rate ISDN Router
centreCOM-AR390Router	{ bridgeRouter 21 }	AT-AR390 E1/G.703 Unchannelised ISDN Router
centreCOM-AR370URouter	{ bridgeRouter 22 }	AT-AR370 Synchronous/Basic Rate ISDN (U interface) Router

The AT Router Sub-tree

The AT Router sub-tree contains a set of objects for managing the AR Router family of multiprotocol routers. Objects have the identifier prefix of *atRouter* ({ alliedTelesyn mibObject brouterMib 4 }). Objects are arranged into six groups (Table C-3 on page C-4).

Table C-3: Object groups in the AT Router sub-tree of the Allied Telesyn Enterprise MIB.

Group	Object Identifier (OID)
mibObject	{ alliedTelesyn 8 }
brouterMib	{ mibObject 4 }
atRouter	{brouterMib 4 }
objects	{ atRouter 1 }
traps	{ atRouter 2 }
sysinfo	{ atRouter 3 }
modules	{ atRouter 4 }
arInterfaces	{ atRouter 5 }
protocols	{ atRouter 6 }

The Objects Group

The Objects Group contains four sets of object identifiers for boards (Table C-4 on page C-5), releases ({ objects 2 }), interface types (Table C-5 on page C-5) and chip sets (Table C-6 on page C-6). These object identifiers are for use with the *hrDeviceID* object in the Host Resources MIB on page C-15.

Table C-4: Object identifiers for AR Router base CPU and expansion boards.

Object	Object Identifier	Description
boards	{ objects 1 }	
ppr_icm_ar023	{ boards 39 }	AT-AR023 SYN1 PRM card
ppr_icm_ar021	{ boards 40 }	AT-AR021 BRI1 PRM card
ppr_icm_ar022	{ boards 41 }	AT-AR022 ETH1 PRM card
ppr_icm_ar025	{ boards 45 }	AT-AR025 E1 PRI1 PRM card
ppr_icm_ar024	{ boards 46 }	AT-AR024 ASYN4 PRM card
ppr_ar300	{ boards 49 }	AT-AR300 (1 BRI S/T ISDN port, 2 POTS ports)
ppr_ar300L	{ boards 52 }	AT-AR300L (1 BRI S/T ISDN port)
ppr_ar310	{ boards 53 }	AT-AR310 (1 BRI S/T ISDN port, 4 POTS ports)
ppr_ar300Lu	{ boards 55 }	AT-AR300LU (1 BRI U ISDN port)
ppr_ar300u	{ boards 56 }	AT-AR300U (1 BRI U ISDN port, 2 POTS ports)
ppr_ar310u	{ boards 57 }	AT-AR310U (1 BRI U ISDN port, 4 POTS ports)
ppr_ar350	{ boards 58 }	AT-AR350 (1 Sync port)
ppr_ar720	{ boards 63 }	AT-AR720 (2 PIC card slots)
ppr_ar010	{ boards 67 }	AT-AR010 EMAC Encryption MAC card
ppr_ar012	{ boards 68 }	AT-AR012 CMAC Compression MAC card
ppr_ar011	{ boards 69 }	AT-AR011 CEMAC Encryption/Compression MAC card
ppr_ar370	{ boards 70 }	AT-AR370 (1 Sync port, 1 BRI S/T ISDN port)
ppr_ar330	{ boards 71 }	AT-AR330 (1 Sync port, 2 LAN ports)
ppr_ar395	{ boards 72 }	AT-AR395 (1 G.703/PRI ISDN port)
ppr_ar390	{ boards 73 }	AT-AR390 (1 G.703/PRI unchannelised port)
ppr_ar370u	{ boards 75 }	AT-AR370U (1 Sync port, 1 BRI U ISDN port)
ppr_icm_ar020	{ boards 76 }	AT-AR020 T1 PRI1 PIC card

Table C-5: Object identifiers for AR Router interface types.

Object	Object Identifier	Description
iftypes	{ objects 3 }	
iface_eth	{ iftypes 1 }	Ethernet
iface_syn	{ iftypes 2 }	Synchronous
iface_asyn	{ iftypes 3 }	Asynchronous
iface_bri	{ iftypes 4 }	BRI ISDN
iface_pri	{ iftypes 5 }	PRI ISDN
iface_pots	{ iftypes 6 }	POTS (voice)

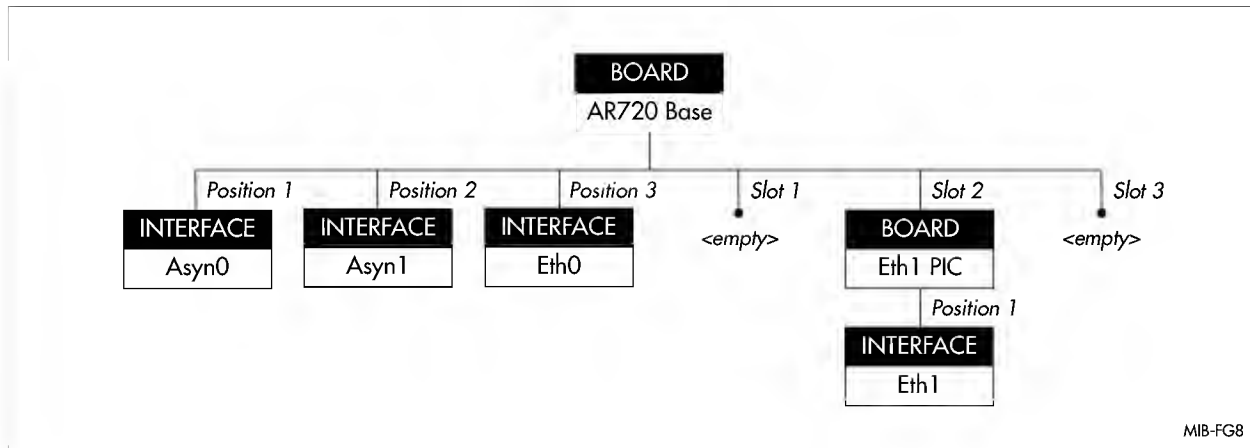
Table C-6: Object identifiers for AR Router chip sets.

Object	Object Identifier	Description
chips	{ objects 4 }	
chip_68020_cpu	{ chips 1 }	68020 processor
chip_68340_cpu	{ chips 2 }	68340 processor
chip_68302_cpu	{ chips 3 }	68302 processor
chip_68360_cpu	{ chips 4 }	68360 processor
chip_860T_cpu	{ chips 5 }	MPC 860T processor
chip_rtc1	{ chips 21 }	Real Time Clock
chip_rtc2	{ chips 22 }	Real Time Clock
chip_rtc3	{ chips 23 }	Real Time Clock
chip_rtc4	{ chips 24 }	Real Time Clock
chip_ram_1mb	{ chips 31 }	1 MB RAM
chip_ram_2mb	{ chips 32 }	2 MB RAM
chip_ram_3mb	{ chips 33 }	3 MB RAM
chip_ram_4mb	{ chips 34 }	4 MB RAM
chip_ram_6mb	{ chips 36 }	6 MB RAM
chip_ram_8mb	{ chips 38 }	8 MB RAM
chip_ram_12mb	{ chips 42 }	12 MB RAM
chip_ram_16mb	{ chips 46 }	16 MB RAM
chip_ram_20mb	{ chips 50 }	20 MB RAM
chip_ram_32mb	{ chips 62 }	32 MB RAM
chip_flash_1mb	{ chips 71 }	1 MB FLASH memory
chip_flash_2mb	{ chips 72 }	2 MB FLASH memory
chip_flash_3mb	{ chips 73 }	3 MB FLASH memory
chip_flash_4mb	{ chips 74 }	4 MB FLASH memory
chip_flash_6mb	{ chips 76 }	6 MB FLASH memory
chip_flash_8mb	{ chips 78 }	8 MB FLASH memory
chip_pem	{ chips 120 }	Processor Enhancement Module

The arInterfaces Group

The arInterfaces Group contains objects that describe the boards, slots and physical interfaces in the router. A router consists of a number of “boards”. Each board may have a number of “positions”, each of which contains a single physical interface. Each board may also have a number of “slots”, which are places which can take other boards. Thus the physical construction of a router may be seen as a tree whose nodes are boards and interfaces, and whose links are positions and slots. Figure Figure C-2 on page C-7 illustrates the object tree for an AR720 with two asynchronous ports and an Ethernet port on the base CPU board, and a single Ethernet PIC in Bay 1.

Figure C-2: The arInterfaces Group object tree for an AR720 with an Ethernet PIC in Expansion Bay 1.



The function of the interface MIB tables is to represent this tree and to map elements in this tree to other MIB variables. Note that any given board has a fixed configuration of positions and slots; it is what is contained in the slots that give different hardware configurations.

The arInterfaces Group contains the following objects:

- *arBoardMaxIndex* is the maximum index of boards in *arBoardTable*. Boards will have indices from 1 to the value of this object. There may be gaps in the sequence if the router has hot-swap capability. If the router has no hot-swap capability, or no swapping has taken place since boot, the sequence of boards will have no gaps. Index 1 is reserved for the main system board of the router.
- *arBoardTable* is a table of the boards in the router, indexed by board index (*arBoardIndex*). This table describes all of the physical boards present in the router. A board is defined as a separately removable circuit board with its own serial number.
- *arSlotTable* is a table of the slots in the router. This table is indexed by board index (*arSlotBoardIndex*) and slot index (*arSlotSlotIndex*) and gives the board index of the board occupying the given slot in the given board.
- *arInterfaceTable* is a table of the physical interfaces in the router. This table is indexed by board index (*arInterfaceBoardIndex*) and a board position (*arInterfacePosition*), and has an entry for each physical interface on the router.

The Modules Group

The Modules Group contains objects that describe particular software modules in the router that are not covered by standard MIBs.

The Ethernet Group

The Ethernet Group has the object identifier prefix *ethernet* ({ modules 23 }), and contains the following objects that describe Ethernet interface(s) on the router:

- *ethIntTable* is a table of the Ethernet interfaces on the router. Each entry is a single Ethernet interface on the router and lists the *ifIndex* of the interface, the index in *arBoardTable* of the board on which the interface resides, the position of the interface on the board, and the duplex mode of the interface.

The ISDN Call Control Group

The ISDN Call Control Group contains objects that describe ISDN call definitions, active call details and call history on the router, and has the object identifier prefix *cc* ({ modules 37 }).

The following objects are defined:

- *ccDetailsTable* is a table of call details. Each entry contains the configuration for a single ISDN call.
- *ccCliListTable* is a table of all the CLI numbers from all CLI lists on the router, indexed by CLI list number. Each entry contains a single CLI number for matching against CLI information.
- *ccActiveCallTable* is a table of the active ISDN calls. Each entry contains the details for a single active call. Since active calls are created by internal router processes, this table is read-only.
- *ccCallLogTable* is a table of ISDN call log entries. Each entry contains log details of a single ISDN call. This table is read-only.
- *ccAttachmentTable* is a table of call detail attachment details, indexed by call detail index and list index of all the attachments from the user module. Each entry lists the index of the call details entry, the index of the attachment, the index of the active call for the attachment and the instance of the user module attached to the ISDN call. Since attachments are generated internally by router processes, this table is read-only.
- *ccBchannelTable* is a table of B channel attachment details, indexed by ISDN interface *ifIndex* and B channel index of all the attachments to ISDN B channels. Each entry lists the B channel *ifIndex*, the B channel number, whether or not the channel is allocated to a call, the type of call (if any), the index of the active call (if any), the priority of the call (if any), and the direction of the call (if any). Since attachments are generated internally by router processes, this table is read-only.

The BRI Group

The BRI Group contains objects that describe BRI interfaces on the router, and has the object identifier prefix *bri* ({ modules 41 }).

The following objects are defined:

- *briIntTable* is a table of the BRI interfaces on the router, indexed by *ifIndex*. Each entry is a single BRI interface on the router and lists the *ifIndex* of the interface, the index in *arBoardTable* of the board on which the interface resides, the position of the interface on the board, the operational mode of the interface (ISDN, TDM or mixed), a bit map of the channels in the interface dedicated to TDM and a bit map of the channels in the interface dedicated to ISDN.
- *briChanTable* is a table of the channels on BRI interfaces. Each entry is a single channel and lists the *ifIndex* of the interface, the channel index, the operational mode of the channel (ISDN, TDM or mixed), and the state (active or inactive) of the channel.

The PRI Group

The PRI Group contains objects that describe PRI interfaces on the router, and has the object identifier prefix *pri* ({ modules 42 }).

The following objects are defined:

- *priIntTable* is a table of the PRI interfaces on the router, indexed by *ifIndex*. Each entry is a single PRI interface on the router and lists the *ifIndex* of the interface, the index in *arBoardTable* of the board on which the interface resides, the position of the interface on the board, the operational mode of the interface (ISDN, TDM or mixed), a bit map of the channels in the interface dedicated to TDM, a bit map of the channels in the interface dedicated to ISDN, and the interface type (E1 or T1).
- *priChanTable* is a table of the channels on PRI interfaces. Each entry is a single channel and lists the *ifIndex* of the interface, the channel index, the operational mode of the channel (ISDN, TDM or mixed), and the state (active or inactive) of the channel.

The Loader Group

The Loader Group contains objects for managing the LOAD module which uses TFTP to download releases, patches, configuration scripts and other files from a TFTP server to NVS or FLASH storage in the router. Objects in this group have the object identifier prefix *loader* ({ modules 48 }).

The following objects are defined:

- *loadTable* is a table of load parameters. There are two entries, one for statically configured load information and one for dynamically configured load information. The static information will be used if there is no dynamic information available. Each entry lists the IP address of the TFTP server, the destination (NVS or FLASH), the name of the file to load, and the delay before loading.
- *loadStatus* is the status of the loader, and is used to start and reset a load, or to report on the progress of a load.

The Install Group

The Install Group contains objects for managing the INSTALL module which controls the software release and patch running on the router. Objects in this group have the object identifier prefix *install* ({ modules 49 }).

The following objects are defined:

- *installTable* is a table of install configurations and controls the software release and patch running on the router. Each entry contains a single install configuration and lists the install type (temporary, preferred or default), the device on which the release is stored (EPROM or FLASH), the release file name, the release version, the device on which the patch is stored (NVS or FLASH), the patch file name and the patch version.
- *installHistoryTable* is a table of descriptions of events in the install history of the router. Each entry is a descriptive line that tells of part of the install history of the last router reboot.
- *configFile* is the name of the file that the router will configure from at boot. If the configuration file name is a zero length string, then there is no configuration file defined in the router.

- *licenceTable* is a table of release licences in the router. This table contains licences for releases of router software stored in FLASH, and lists the licence index, licence status, licenced software, version, licence password and licence expiry date.
- *createConfigFile* is the name of a file to create containing the current router configuration. A read from this variable will return the same as the variable *configFile*. To save the current configuration in the current configuration file, read *createConfigFile* first, then write the result back to *createConfigFile*. If this variable is written with the name of an existing file, the file will be replaced with the current configuration.

The File Group

The File Group contains objects for managing the file system in the router. Objects in this group have the object identifier prefix *file* ({ modules 56 }).

The following objects are defined:

- *fileTable* is a table of all the files in the router's non-volatile storage. Each entry contains the details of a single file and lists the file name, the device on which the file is stored (FLASH or NVS), the size of the file in bytes, the date and time the file was created, and the status of the file.

The Firewall Group

The Firewall Group contains objects describing traps generated by the firewall in the router. Objects in this group have the object identifier prefix *firewall* ({ modules 77 }).

The following objects are defined:

- *firewallTrapMessage* is the last TRAP message sent from the firewall. This variable is really just a placeholder for the object sent in the firewall TRAP, but can be read independently if required. Note however that a new firewall TRAP will cause this variable to be overwritten.
- *firewallTrap* is a trap message generated when the firewall detects an intrusion or attack. Firewall trap notifications are enabled using the command:

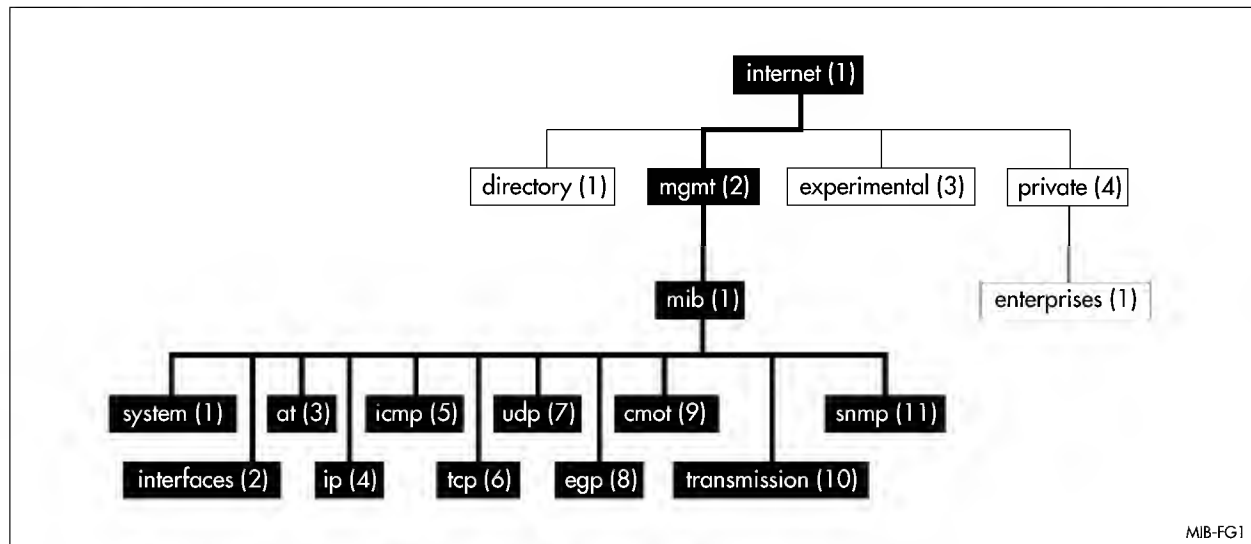
```
ENABLE FIREWALL NOTIFY=SNMP
```

MIB-II MIB

RFC 1213, “*Management Information Base for Network Management of TCP/IP-based internets: MIB-II*” defines the core set of managed objects for TCP/IP-based internets, and supersedes MIB-I defined in RFC 1156.

Objects defined in MIB-II reside in the `mib(1)` sub-tree (Figure C-3 on page C-11) and have the object identifier prefix `mib-2` ({`mgmt 1`}).

Figure C-3: The MIB-II sub-tree of the Internet-standard Management Information Base (MIB).



The objects in MIB-II are arranged into 11 groups:

- The *System* group contains objects that describe contact, administrative, location, and service information for the entity.
- The *Interfaces* group contains objects that describe the interfaces on the entity. Each interface is thought of as being attached to a 'subnetwork'. Note that this term should not be confused with 'subnet' which refers to an addressing partitioning scheme used in the Internet suite of protocols.
- The *Address Translation* group contains objects that describe the translation between network addresses (e.g. IP addresses) and subnetwork-specific, or physical, addresses on the entity.
- The *IP* group contains objects that describe the entity's IP addressing scheme, IP routing table, IP address translation and counters for the IP forwarding process.
- The *ICMP* group contains objects that record traffic statistics for the ICMP protocol on the entity.
- The *TCP* group contains objects that describe the TCP protocol, active TCP connections and TCP traffic counters on the entity.
- The *UDP* group contains objects that describe the UDP protocol, active UDP connections and UDP traffic counters on the entity.
- The *EGP* group contains objects that describe the entity's EGP neighbours and traffic counters for the EGP protocol on the entity.
- The *CMOT* group is preserved for historical reasons and is not used for management of the entity.

- The *Transmission* group contains objects that describe the different types of transmission media supported on the entity.
- The *SNMP* group contains objects that describe the SNMP protocol on the entity and traffic counters for the SNMP protocol on the entity.

These groups are the basic unit of conformance: if any of the objects in a group are applicable to an implementation, then it must implement all objects in that group.

Implementation

All router models implement all groups in MIB-II. However, the implementation of some objects differs from RFC 1213 (Table C-7 on page C-12). In particular, some read-write objects are implemented as read-only.

Table C-7: MIB-II implementation variations.

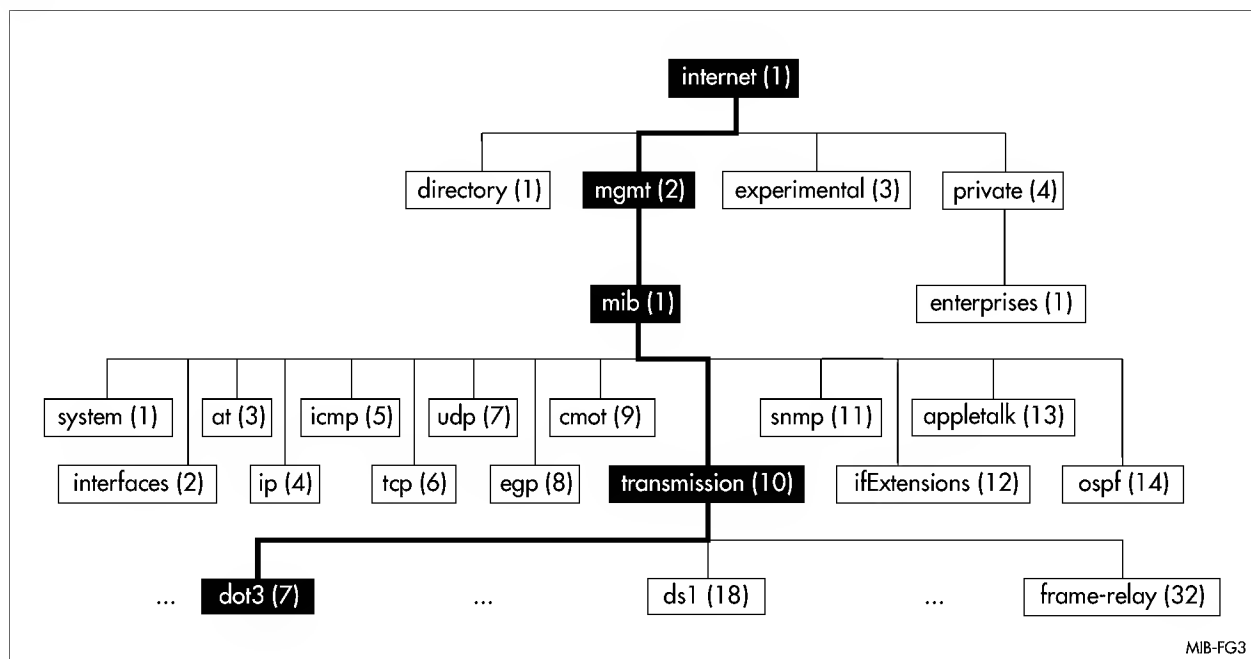
Object Name	Object ID	Access	Implementation
ifAdminStatus	{ 1.3.6.1.2.1.2.2.1.7 }	Read-write	Read-write for PPP & Syn interfaces Read-only for all other interfaces
atIfIndex	{ 1.3.6.1.2.1.3.1.1.1 }	Read-write	Read-only
atPhysAddress	{ 1.3.6.1.2.1.3.1.1.2 }	Read-write	Read-only
atNetAddress	{ 1.3.6.1.2.1.3.1.1.3 }	Read-write	Read-only
ipRouteIfIndex	{ 1.3.6.1.2.1.4.21.1.2 }	Read-write	Read-only
ipRouteMetric1	{ 1.3.6.1.2.1.4.21.1.3 }	Read-write	Read-only
ipRouteMetric2	{ 1.3.6.1.2.1.4.21.1.4 }	Read-write	Read-only
ipRouteMetric3	{ 1.3.6.1.2.1.4.21.1.5 }	Read-write	Read-only
ipRouteMetric4	{ 1.3.6.1.2.1.4.21.1.6 }	Read-write	Read-only
ipRouteNextHop	{ 1.3.6.1.2.1.4.21.1.7 }	Read-write	Read-only
ipRouteType	{ 1.3.6.1.2.1.4.21.1.8 }	Read-write	Read-only
ipRouteAge	{ 1.3.6.1.2.1.4.21.1.10 }	Read-write	Read-only
ipRouteMask	{ 1.3.6.1.2.1.4.21.1.11 }	Read-write	Read-only
ipRouteMetric5	{ 1.3.6.1.2.1.4.21.1.12 }	Read-write	Read-only
ipNetToMediaIfIndex	{ 1.3.6.1.2.1.4.22.1.1 }	Read-write	Read-only
ipNetToMediaPhysAddress	{ 1.3.6.1.2.1.4.22.1.2 }	Read-write	Read-only
ipNetToMediaNetAddress	{ 1.3.6.1.2.1.4.22.1.3 }	Read-write	Read-only
ipNetToMediaType	{ 1.3.6.1.2.1.4.22.1.4 }	Read-write	Read-only
tcpConnState	{ 1.3.6.1.2.1.6.13.1.1 }	Read-write	Read-only
egpNeighEventTrigger	{ 1.3.6.1.2.1.8.5.1.15 }	Read-write	Read-only
snmpEnableAuthenTraps	{ 1.3.6.1.2.1.11.30 }	Read-write	Read-only

Ethernet-like Interface Types MIB

RFC 1398, “Definitions of Managed Objects for the Ethernet-like Interface Types” defines a portion of the Management Information Base (MIB) for use with network management protocols in TCP/IP-based internets. In particular, it defines objects for managing Ethernet-like objects.

Objects defined in this MIB reside in the `mib(1)` sub-tree, under the Transmission Group defined in MIB-II (Figure C-4 on page C-13) and have the object identifier prefix `dot3` ({transmission 7}).

Figure C-4: The Ethernet-like interface types sub-tree of the Internet-standard Management Information Base (MIB).



Instances of these object types represent attributes of Ethernet-like interfaces. At present, Ethernet-like media are identified by three values of the *ifType* object in the Internet Standard MIB:

- ethernet-csmacd(6)
- iso88023-csmacd(7)
- starLan(11)

For these interfaces, the value of the *ifSpecific* variable in MIB-II has the object identifier value `dot3` ({transmission 7}).

The objects in this MIB are organised into two groups:

- The *Ethernet-like Statistics* group contains objects that record statistics for Ethernet-like interfaces on the entity.
- The *Ethernet-like Collision Statistics* group contains objects that describe collision histograms for Ethernet-like interfaces on the entity.

Implementation

All router models implement all objects in the Ethernet-like Interface Types MIB.

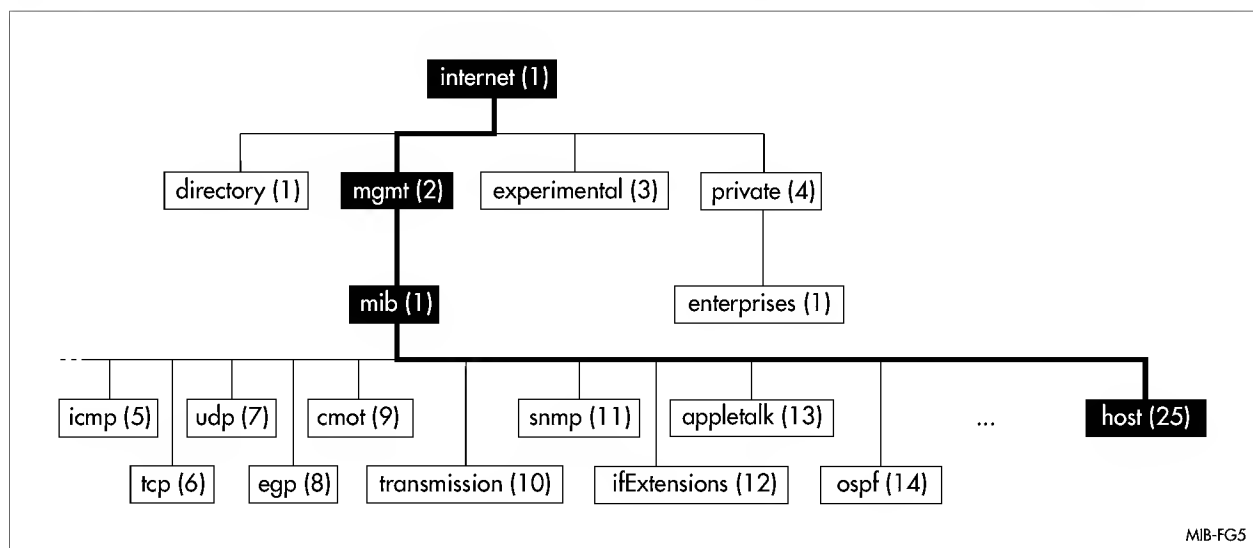
Host Resources MIB

RFC 1514, “*Host Resources MIB*” defines a portion of the Management Information Base (MIB) for use with network management protocols in TCP/IP-based internets. In particular, it defines objects for managing host systems.

The term host is construed to mean any computer that communicates with other similar computers attached to the internet and that is directly used by one or more human beings. Although this MIB does not necessarily apply to devices whose primary function is communications services (e.g. terminal servers, routers, bridges, monitoring equipment), such relevance is not explicitly precluded. This MIB instruments attributes common to all internet hosts including, for example, both personal computers and systems that run variants of Unix.

Objects defined in this MIB reside in the `mib(1)` sub-tree (Figure C-5 on page C-14) and have the object identifier prefix `host` ({`mib-2 25`}).

Figure C-5: The Host Resources sub-tree of the Internet-standard Management Information Base (MIB).



The MIB is organised into six groups:

- The *Host Resources System* group contains objects that describe general system configuration parameters.
- The *Host Resources Storage* group contains objects that describe the logical storage areas on the host.
- The *Host Resources Device* group contains objects that describe the devices on the host.
- The *Host Resources Running Software* group contains objects that describe the software that is running or loaded into physical or virtual memory in preparation for running, including the host's operating system, device drivers, and applications.
- The *Host Resources Running Software Performance* group contains objects that describe performance metrics for each entry in the *hrSWRunTable*.
- The *Host Resources Installed Software* group contains objects that describe each piece of software installed in long-term storage (e.g. a disk drive) locally on this host. This does not include software loadable remotely from a network server.

Implementation

All router models implement the following groups in the Host Resources MIB:

- The *Host Resources System* group.
- The *Host Resources Storage* group.
- The *Host Resources Device* group.

However, the implementation of some objects differs from RFC 1514 (Table C-8 on page C-15).

The following groups in the Host Resources MIB are not implemented because they are not meaningful in the context of the router:

- The *Host Resources Running Software* group
- The *Host Resources Running Software Performance* group
- The *Host Resources Installed Software* group

Table C-8: Host Resources MIB implementation variations.

Object Name	Object ID	Access	Implementation
hrSystemDate	{ 1.3.6.1.2.1.25.1.2 }	Read-write	Read-only.
hrSystemInitialLoadDevice	{ 1.3.6.1.2.1.25.1.3 }	Read-write	The index of the <i>hrDeviceEntry</i> for the device (NVS or FLASH) containing the system boot script. Read-only.
hrSystemInitialLoadParameters	{ 1.3.6.1.2.1.25.1.4 }	Read-write	The name of the system boot script (as set by the SET CONFIG command). Read-only.
hrSystemNumUsers	{ 1.3.6.1.2.1.25.1.5 }	Read-only	The number of active user (Telnet and asynchronous) sessions that have been authenticated via the USER module.
hrSystemProcesses	{ 1.3.6.1.2.1.25.1.6 }	Read-only	Always returns the value 1.
hrSystemMaxProcesses	{ 1.3.6.1.2.1.25.1.7 }	Read-only	Always returns the value 1.
hrStorageAllocationUnits	{ 1.3.6.1.2.1.25.2.3.1.4 }	Read-only	Returns the value 2048 for RAM buffers, or 1 for NVS and FLASH.
hrDeviceType	{ 1.3.6.1.2.1.25.3.2.1.2 }	Read-only	Table C-9 on page C-16 lists the devices supported by the router.
hrDeviceID	{ 1.3.6.1.2.1.25.3.2.1.4 }	Read-only	Object identifier values are from the Objects Group of the Enterprise MIB on page C-4.
hrNetworkTable	{ 1.3.6.1.2.1.25.3.4 }	-	Not implemented.
hrPrinterTable	{ 1.3.6.1.2.1.25.3.5 }	-	Not implemented. The router provides print services (e.g. LPD) but it is not concerned with issues such as printer diagnosis which are required to implement this portion of the MIB.
hrDiskStorageTable	{ 1.3.6.1.2.1.25.3.6 }	-	Not implemented because the router does not have any entries in the device table of type <i>hrDeviceDiskStorage</i> . FLASH and NVS are represented as raw non-volatile memory.
hrFSTable	{ 1.3.6.1.2.1.25.3.8 }	-	Not implemented because the router does not support any file systems close enough in semantics to those catered for by this MIB.
Host Resources Running Software Group	-	-	Not implemented on the router because it is not meaningful in the context of the router.

Table C-8: Host Resources MIB implementation variations.

Object Name	Object ID	Access	Implementation
Host Resources Running Software Performance Group	-	-	Not implemented on the router because it is not meaningful in the context of the router.
Host Resources Installed Software Group	-	-	Not implemented on the router because it is not meaningful in the context of the router.

Table C-9: Host Resources MIB device types supported by the router.

hrDeviceType	Router Device
hrDeviceProcessor	Processor
hrDeviceNetwork	LAN/WAN network port
hrDeviceCoprocessor	MAC card
hrDeviceSerialPort	Asynchronous ports
hrDeviceClock	RTC
hrDeviceVolatileMemory	RAM, FSRAM, CAM
hrDeviceNonVolatileMemory	NVS, FLASH

Glossary

Symbols

10BASET 10 Mbps/baseband/twisted pair. The IEEE standard for twisted pair Ethernet.

802.2 The IEEE standard for the definition of the Logical Link Control protocol for LANs.

802.3 The IEEE standard for the definition of the CSMA/CD (Ethernet) medium access method for LANs.

A

abuse of privilege When a user performs an action that they should not have, according to organizational policy or law.

ACK *Acknowledgement.* A packet sent to indicate that a block of data arrived at its destination without error. For example, at the link level, an acknowledgement indicates successful transmission across a single hardware link; at the transport level an acknowledgement indicates successful transmission between end systems (possibly over multiple hardware links). See NAK.

address mask See subnet mask.

address resolution Conversion of a network layer address into a corresponding physical address. Depending on the underlying network, address resolution may require broadcast on a local network. See ARP, RARP.

address resolution protocol See ARP.

aging The process applied to a routing table whereby entries are aged to prevent the database filling up with entries that are no longer valid. If an entry reaches a specified maximum age before an update is received for that entry, the entry is no longer used in routing decisions and is eventually removed from the database.

American National Standards Institute See ANSI.

anonymous FTP Anonymous FTP allows a user to retrieve documents, files, programs, and other archived data from anywhere in the Internet without



having to establish a user ID and password. By using the special user ID of anonymous the network user will bypass local security checks and will have access to publicly accessible files on the remote system. See archive site, FTP.

ANSI *American National Standards Institute*. An organisation responsible for coordinating and approving U.S. standards. Standards approved by ANSI are often called ANSI standards. ANSI is the U.S. representative to ISO.

application-level firewall A firewall system in which service is provided by processes that maintain complete TCP connection state and sequencing. Application level firewalls often re-address traffic so that outgoing traffic appears to have originated from the firewall, rather than from the internal host.

archie A system to automatically gather, index and serve information on the Internet. The initial implementation of archie provided an indexed directory of file names from all anonymous FTP archives on the Internet. Later versions provide other collections of information. See archive site.

archive site A machine that provides access to a collection of files across the Internet. An "anonymous FTP archive site", for example, provides access to this material via the FTP protocol. See anonymous FTP, archie.

ARP *Address Resolution Protocol*. The TCP/IP protocol used to dynamically map a high level IP address to a low-level physical (hardware) address on a local area network. ARP applies only across a single physical network and is limited to networks that support hardware broadcast. See address resolution, proxy ARP, RARP.

AS See autonomous system.

ASCII *American Standard Code for Information Interchange*. A standard character-to-number encoding widely used in the computer industry.

assigned A term used with the router to refer to the state of an asynchronous port that currently has a logical connection to a service on the network.

Assigned Numbers A set of values (usually numeric) used by TCP/IP protocols. They are documented in a number of RFCs, the most recent being RFC 1340. See RFC.

asynchronous Transmission in which each character is sent individually. The time intervals between transmitted characters may be of unequal length. Transmission is controlled by *start* and *stop* elements before and after each character.

attention character A term used with the router to refer to a special character (either [Break] or [Ctrl/P]) that signals to the router that the next character is a function character and should be sent to the terminal server software, not the remote process to which the user is assigned. Function characters signal to the router that a special action is to be taken.

authentication The property of knowing the actual sender of a message, and knowing that the message has not been interfered with in transit; and the procedure employed with the aim of confirming the identity of the sender. Authentication can be subdivided into weak and strong authentication. Traditional reusable passwords are considered weak authentication because, if compromised, they can be used to repeatedly gain access to a host. Stronger authentication methods are normally based on cryptographic techniques and



often rely on the authorised user knowing something (like a password or passphrase) and having something (like a key, or a hardware token).

authentication token A portable device used for authenticating a user. Authentication tokens operate by challenge/response, time-based code sequences, or other techniques. This may include paper-based lists of one-time passwords.

authorisation The process of determining what types of activities a user is permitted to undertake. Usually, authorization is in the context of authentication: once you have authenticated a user, they may be authorized for different types of access or activity.

autobauding A operational mode of the terminal server software in the router, in which the router automatically adjusts the speed of an asynchronous port to match the speed of the terminal connected to the port.

autonomous system A collection of gateways or routers under one administrative entity using a common interior gateway protocol (IGP). Gateways and routers within an autonomous system have a high degree of trust.

B

backbone The primary connectivity mechanism of a hierarchical distributed system. All systems which have connectivity to an intermediate system on the backbone are assured of connectivity to each other. This does not prevent systems from setting up private arrangements with each other to bypass the backbone for reasons of cost, performance, or security.

BACP *Bandwidth Allocation Control Protocol*. A PPP NCP used to negotiate the use of BAP on a multilink PPP interface. BACP is an IETF standard defined in RFC 2125.

bandwidth Technically, the difference, in Hertz (Hz), between the highest and lowest frequencies of a transmission channel. However, as typically used, the amount of data that can be sent through a given communications circuit. For example, Ethernet has a bandwidth of 10Mbps.

BAP *Bandwidth Allocation Protocol*. An IETF protocol, defined in RFC 2125, that provides a mechanism for two PPP peers to manage the bandwidth available to the protocols using a multilink PPP bundle by negotiating gracefully to add and remove links from the multilink bundle.

Basic Rate Access A mode of access to an ISDN service that provides two 64 kbit/s B channels for data and one 16 kbit/s D channel for link control and management. The access point is normally at a customer's premises. See BRI, ISDN.

bastion host A system that has been hardened to resist attack, and which is installed on a network in such a way that it is expected to potentially come under attack. Bastion hosts are often components of firewalls, or may be "outside" Web servers or public access systems. Generally, a bastion host is running some form of general purpose operating system (e.g., UNIX, VMS, Windows NT, etc.) rather than a ROM-based or firmware operating system.



baud Literally, the number of times per second the signal can change on a transmission line. It is normally equal to the number of bits per second that can be transferred. The underlying transmission system may use some of the bandwidth. For asynchronous lines, the number of characters per second that can be transmitted is estimated by dividing the baud rate by ten.

best-effort delivery Characteristic of network technologies that do not provide reliability at link levels. The combination of IP and UDP protocols provides a best-effort delivery service to applications.

BIA *Best Information Algorithm*. An algorithm, similar to split horizon, used to determine which interfaces to send routing information broadcasts to, and what the broadcasts should contain.

boot A term used in computing to refer to the process of starting a computer, loading the operating system or executive program from disk or ROM.

bps *bits per second*. A measure of the rate of data transmission.

BRI *Basic Rate Interface*. In the router, the name of the software module, and the name assigned to logical interfaces, that provide Basic Rate Access to an ISDN service. See Basic Rate Access, ISDN.

bridge A device that connects two or more networks and forwards packets between them. Bridges normally operate at the MAC level, for example connecting Ethernets. Bridges can usually be configured to filter packets; that is, to forward only certain traffic. Bridges differ from repeaters and routers in that bridges store and forward complete packets, whereas repeaters simply forward re-timed electrical signals from one cable to another. See repeater, router.

broadcast A packet delivery system that delivers a copy of a given packet to all hosts attached to the network. For example, Ethernet. See directed broadcast, multicast, unicast.

BSD *Berkeley Software Distribution*. Implementation of the UNIX operating system and its utilities developed and distributed by the University of California at Berkeley. "BSD" is usually preceded by the version number of the distribution, e.g., "4.3BSD".

buffer A block of memory used to store data temporarily.

bundle A number of active PPP links, with a common peer, grouped together as a single PPP link using the multilink procedure as defined in RFC 1990.

C

CCITT *International Consultative Committee for Telegraphy and Telephony*. A unit of the International Telecommunications Union (ITU) of the United Nations. CCITT sets standards, known as "Recommendations," for all internationally controlled aspects of analog and digital communications. For example, CCITT defined the X.25 network protocols.

CCP *Compression Control Protocol*. A PPP NCP used to negotiate the use of compression on a PPP interface. CCP is an IETF standard defined in RFC 1962.

CD *Carrier Detect*. A modem control line which is an input to the router from a modem or NTU signifying that the modem or NTU is receiving a valid carrier



signal. CD asserted at both ends of a data link is an indication that the link is operational.

challenge/response An authentication technique whereby a server sends an unpredictable challenge to the user, who computes a response using some form of authentication token.

checksum A small, integer value computed from a sequence of octets by treating them as integers and computing the sum. A checksum is used to detect transmission errors. The sender computes a checksum and appends it to a packet when transmitting. The receiver verifies the packet's contents by re-computing the checksum and comparing it to the value sent. Many TCP/IP protocols use a 16-bit checksum computed with one's complement arithmetic.

circuit A term used in networking to refer to a logical stream of data between two users in the network. A single physical link may have several circuits running on it.

coaxial cable An electrical cable in which a piece of wire is surrounded by insulation and a tubular conductor (or mesh) whose axis of curvature coincides with the center of the piece of wire, hence the term "coaxial". Examples include thick- and thinwire Ethernet.

compression A technique for reducing the apparent amount of traffic on a data link. The router, for instance, supports Van Jacobson's header compression for IP over Point-to-Point Protocol links. This is an option which reduces the normal 40 byte header to 4–5 bytes.

congestion A condition that occurs when the offered load exceeds the capacity of a data communication path.

connectionless A model of interconnection in which communication takes place without first establishing a connection. It is sometimes called datagram. Each packet of data is treated as a separate entity containing a source and destination address. Usually, connectionless services can drop packets or deliver them out of sequence. Packets may also take different routes to the same destination. Examples include LANs, IP and UDP.

connection-oriented The model of interconnection in which communication proceeds through three well-defined phases: connection establishment, data transfer, connection release. Examples are X.25 and TCP.

CRC *Cyclic Redundancy Check*. A method of checking the integrity of received data, using a polynomial algorithm based on the content of the data. The term is also used to refer to the computed value. The sender computes a CRC and appends it to a packet when transmitting. The receiver verifies the packet's contents by re-computing the CRC and comparing it to the value sent. CRCs are more expensive to compute than a checksum, but can detect more errors.

CPU *Central Processing Unit*. In the router, this is a microprocessor that controls all operations necessary to the functioning of the router.

CSMA/CD *Carrier Sense Multiple Access with Collision Detection*. The access method used by local area networking technologies such as Ethernet. Multiple stations contend for access to a transmission medium by listening to see if it is idle. A mechanism is provided to detect when two stations simultaneously attempt to transmit data.



D

daemon A UNIX term referring to a process that is not connected with a user but performs a service, such as a mail daemon or an FTP server daemon.

data driven attack A form of attack in which the attack is encoded in innocuous-seeming data which is executed by a user or other software to implement an attack. In the case of firewalls, a data driven attack is a concern since it may get through the firewall in data form and launch an attack against a system behind the firewall.

data link layer The network layer that is responsible for data transfer across a single physical connection, or series of bridged connections, between two network entities.

datagram A self-contained, independent entity of data carrying sufficient information to be routed from the source to the destination computer without reliance on earlier exchanges between this source and destination computer and the transporting network. See frame, packet.

DCE *Data Communication Equipment*. The term applied by X.25 protocols to equipment that forms a packet switching network, to distinguish it from the computers and terminals that connect to the network. See DTE.

default route A routing table entry which is used to direct packets addressed to networks not explicitly listed in the routing table. See route table.

defence in depth The security approach whereby each system on the network is secured to the greatest possible degree.

dialup A temporary, as opposed to dedicated, connection between machines established over a standard phone line.

directed broadcast A packet deliver system that delivers a copy of a given packet to "all hosts" on a specific network. A single copy of a directed broadcast is routed to the specified network where it is broadcast to all machines on that network.

DMAC *Direct Memory Access Controller*. An integrated circuit that mediates the transfer of data between a peripheral, such as an Ethernet controller, and memory, without CPU intervention.

DNS *Domain Name System*. The distributed name/address mechanism used in the Internet. It comprises distributed online databases that contain mappings between human-readable names and IP addresses, and servers which provide translation services to client applications.

DNS spoofing Assuming the DNS name of another system by either corrupting the name service cache of a victim system, or by compromising a domain name server for a valid domain.

domain A part of the DNS naming hierarchy. Syntactically, an Internet domain name consists of a sequence of names (labels) separated by periods (dots), e.g., "machine.company.com". See DNS.

dot address See dotted decimal notation.



dotted decimal notation The syntactic representation for a 32-bit integer that consists of four 8-bit numbers written in base 10 with periods (dots) separating them. It is used to represent IP addresses in the Internet, e.g. 172.16.9.197.

downline loading A term used with the router to refer to the process of transferring a code patch over a TCP/IP link from a TFTP server to the router. It is a mechanism for fixing bugs and adding enhancements to the software used in the router.

DTE *Data Terminal Equipment*. The term applied by X.25 protocols to computers and terminals to distinguish them from the packet switching network to which they connect. See DCE.

DTR *Data Terminal Ready*. An RS-232C electrical signal asserted by the router on a port when it is ready to transmit and receive data on the port.

DVMRP *Distance Vector Multicast Routing Protocol*. A multicast routing protocol.

E

Email *Electronic mail*. A system enabling a computer user to exchange messages with other computer users (or groups of users) via a communications network. Electronic mail is one of the most popular uses of internets.

Email address The address that is used to send electronic mail to a specified destination. For example, "ford.prefect@earth.com".

encapsulation The technique used by layered protocols in which a layer adds header information to the protocol data unit (PDU) from the layer above. As an example, in Internet terminology, a packet would contain a header from the physical layer, followed by a header from the network layer (IP), followed by a header from the transport layer (TCP), followed by the application protocol data.

EPROM *Erasable Programmable Read-Only Memory*. These devices contain the system software on the router, and may need to be changed in some circumstances to upgrade the software to a new release. They are nonvolatile, i.e. they retain their information during power-down. See FLASH

Ethernet A common, 10Mbps local area network technology invented by Xerox Corporation at the Palo Alto Research Center. Ethernet is a best-effort delivery system that uses CSMA/CD technology. Ethernet can be run over thinwire coaxial cable (10BASE2), thickwire coaxial cable (10BASE5), twisted pair cable (10BASET), or fibre optic cable.

exclusion filter A process in which a router discards data from sources whose addresses appears in an exclusion list, rather than forwarding the data to other networks, effectively filtering the data from internetwork traffic. See inclusion filter.

F

FCS *Frame check sequence*. Bytes added to a frame so that the integrity of the frame may be checked. Typically the bytes are a CRC of the data in the frame.

FIFO *First In, First Out*. A first in, first out buffer. FIFO buffers are useful for handling data that arrives in bursts, as the data can be buffered until the computer is able to process it. Data is processed (removed from the buffer) in



the order it was added to the buffer. A contrasting buffer system is LIFO (Last In, First Out), often referred to as a stack.

file transfer The process of copying of a file from one computer to another over a computer network. See anonymous FTP, FTP.

File Transfer Protocol See FTP.

filter Within the router, A filter is a process used to select which packets will be processed by the router, and which will be ignored or discarded. Selection may be based on addresses or protocol type.

firewall A system or combination of systems that enforces a boundary between two or more networks.

flag A program-readable indicator that can be used to signal an event or a state, or provide simple data values (e.g. TRUE/FALSE, ON/OFF, use option X). For example, in the HDLC data link protocol, the bit pattern 01111110 is used to flag the beginning and end of a frame.

FLASH A new memory technology which combines the nonvolatile features of EPROMs with the easy in-system reprogramming of conventional volatile RAM. See EPROM.

flow control Control of the rate at which devices inject packets into a network, usually to avoid congestion. Flow control mechanisms can be implemented in hardware and/or software, at various protocol layers, and with varying complexity. See XON/XOFF.

fragment A piece of a packet. When a router is forwarding an packet to a network that has a maximum packet size smaller than the packet size, it is forced to break up that packet into multiple fragments. These fragments will be reassembled at the destination. See fragmentation, MTU.

fragmentation The process in which a packet is broken into smaller pieces to fit the requirements of a given physical network. The reverse process is termed reassembly. See fragment, MTU.

frame A frame is a data link layer "packet" which contains the header and trailer information required by the physical medium. That is, network layer packets are encapsulated to become frames. See datagram, encapsulation, packet.

FTP *File Transfer Protocol*. The TCP/IP standard, high-level protocol for transferring files from one computer to another over a network. FTP is also usually the name of the program that the user invokes to execute the protocol. See anonymous FTP.

function A special character sent to the terminal server software in the router, rather than the remote process to which the user's terminal is assigned, to indicate that a special action is to be taken. A function is sent to the router by pressing the *attention* character ([Break] or [Ctrl/P]) followed by a function character.



G

gateway The original Internet term for what is now called router or more precisely, IP router. In modern usage, the terms “gateway” and “application gateway” refer to systems which perform translation from some native protocol or physical data format to another. Examples include electronic mail gateways which translate between X.400 and RFC 822 mail message formats. See router.

H

HDLC *High level Data Link Control*. ISO’s data link level protocol for OSI. It was adapted by CCITT for it’s link access protocol (LAP/LAPB) used with X.25 networks..

header The portion of a packet, preceding the actual data, containing source and destination addresses, and error checking and other fields. A header is also the part of an electronic mail message that precedes the body of a message and contains, among other things, the message originator, date and time. See Email, packet.

hello packet Hello packets are used in a number of network protocols, to perform similar functions. Typically, a Hello packet is used to advertise a node’s presence to the network or to establish and maintain information about the presence of other nodes (including hosts and routers) in the network.

HMAC *Hashed Message Authentication Code* A method of generating a message authentication code using a secure hash function such as MD5 or SHA-1. HMAC is defined in RFC2104.

hop count A measure of distance between two points in an internet. A hop count of *n* means that *n* gateways or routers separate the source and destination.

host An (end-user) computer system that connects to a network, such as a PC, minicomputer or mainframe.

host-based security The technique of securing an individual system from attack. Host based security is operating system and version dependent.

I

ICMP *Internet Control Message Protocol*. The TCP/IP protocol used to handle errors and control messages at the IP layer. ICMP is part of the IP protocol. Gateways, routers and hosts use ICMP to send reports of problems about datagrams back to the original source that sent the datagram.

IEEE *Institute of Electrical and Electronics Engineers*. A standard-making body in the U.S. responsible for the 802 standards for local area networks.

IEEE 802.3 See 802.3.

IETF *Internet Engineering Task Force*. One of the task forces of the IAB (*Internet Activities Board*). It is a large, open community of network designers, operators, vendors, and researchers whose purpose is to coordinate the operation, management and evolution of the Internet, and to resolve short-range and mid-range protocol and architectural issues. It is a major source of proposals for protocol standards which are submitted to the IAB for final approval.



IGMP *Internet Group Management Protocol* A protocol for managing the addition and deletion of hosts from multicast groups.

IGP *Interior Gateway Protocol*. The generic term applied to protocols used to exchange routing information between collaborating routers within an autonomous system. RIP and OSPF are examples of IGPs. See RIP.

inclusion filter A process in which a router forwards data to other networks only from sources whose addresses appear in an inclusion list, effectively filtering data from other sources from internetwork traffic. See exclusion filter.

insider attack An attack originating from inside a protected network.

instance A term used in the router to refer to an instantiation of an interface type associated with a particular synchronous port or Ethernet port on the router.

interface One of the physical ports on the router, including the Ethernet and asynchronous ports.

interface type The type (Ethernet or Point-to-Point) of one of the interfaces on the router.

internet A collection of networks interconnected by a set of routers which allow them to function as a single, large virtual network.

Internet (note the capital "I") The largest internet in the world consisting of large national backbone networks (such as MILNET, NSFNET, and CREN) and a myriad of regional and local campus networks all over the world. The Internet is a multiprotocol network, but generally carries TCP/IP.

Internet address See IP address.

Internet Draft Internet Drafts are working documents of the IETF. They are usually precursors to RFCs.

Internet Protocol See IP.

interoperability The ability of software and hardware on multiple machines from multiple vendors to communicate meaningfully.

intrusion detection Detection of break-ins or break-in attempts either manually or via software expert systems that operate on logs or other information available on the network.

IP *Internet Protocol*. The network layer protocol for the TCP/IP protocol suite. It is a connectionless, best-effort packet switching protocol.

IP address A 32-bit address assigned to hosts using TCP/IP. The address specifies a specific connection to a network, not the host itself. See dotted decimal notation.

IP datagram The fundamental unit of information passed across the Internet. It contains a source and destination address along with data and a number of fields which define such things as the length of the datagram, the header checksum, and flags to say whether the datagram can be (or has been) fragmented.



IP interface An entity representing an IP layer attached to a layer 2 interface and all information the IP routing algorithm needs to know to use the layer 2 interface to transmit datagrams over that physical connection. An IP interface consists of one or more IP logical interfaces.

IP logical interface An entity which represents an IP layer interface and holds all network layer specific information such as network address, mask, metric, etc. Multiple logical interfaces can be bundled together in a single IP interface.

IP splicing/hijacking An attack in which an active, established, session is intercepted and co-opted by the attacker. IP splicing attacks may occur after an authentication has been made, permitting the attacker to assume the role of an already authorised user. Primary protections against IP splicing rely on encryption at the session or network layer.

IP spoofing An attack in which a system attempts to illicitly impersonate another system by using its IP network address.

ISDN *Integrated Services Digital Network*. A technology which combines voice and digital network services in a single medium, making it possible for telecommunications providers to offer customers digital data services as well as voice connections through a single "wire". The standards that define ISDN are specified by CCITT.

L

LAN *Local Area Network*. Any physical network technology (such as Ethernet) that operates at high speed (typically 10 Mbits per second or more) over short distances (up to a few kilometres). See WAN.

layer Communication networks for computers may be organized as a set of more or less independent protocols, each in a different layer (also called level). The lowest layer governs direct host-to-host communication between the hardware on different hosts; the highest layer consists of user applications. Each layer builds on the layer beneath it. For each layer, programs at different hosts use protocols appropriate to the layer to communicate with each other. TCP/IP has five layers of protocols; OSI has seven. The advantages of different layers of protocols is that the methods of passing information from one layer to another are specified clearly as part of the protocol suite, and changes within a protocol layer are prevented from affecting the other layers. This greatly simplifies the task of designing and maintaining communication programs.

layer 2 Interface An entity representing the layer 2 interface in the OSI/ISO network layering model, also referred to as a link layer interface. Examples are Ethernet, PPP, and X.25.

LCP *Link Control Protocol*. Part of the Point-to-Point Protocol that establishes and configures a link between the two stations at each end of a point-to-point link.

LED *Light Emitting Diode*. A luminous indicator.

LLC Logical Link Control. The upper portion of the data link layer, as defined in IEEE 802.2. The LLC sublayer presents a uniform interface to the user of the data link service, usually the network layer. Beneath the LLC sublayer is the MAC sublayer.



local interface A default logical interface for all locally generated IP packets.

Logging The process of storing information about events that occurred on the firewall or network.

loopback A state in which data transmitted is also received. Normally it is used to test data links by applying a loopback at various points and verifying successful reception of the data transmitted.

M

MAC *Media Access Control*. The lower portion of the data link layer. The MAC differs for various physical media. See LLC.

MAC address The hardware address of a device connected to a shared media. For example, the MAC address of a PC on an Ethernet is its Ethernet address.

management information base See MIB.

mask A bit pattern used to “mask out” portions of data.

MD5 *Message Digest algorithm 5*. A method of producing a hash, or fingerprint, of a block of data. MD5 is defined in RFC1321.

metric A concept used to describe the cost of a route across a network, the distance to the destination at the remote end of the route, or the capacity of the route.

MIB *Management Information Base*. The set of parameters an SNMP management station can query or set in the SNMP agent of a network device (e.g., router). Standard MIBs have been defined, and vendors can develop private MIBs. In theory, any SNMP manager can talk to any SNMP agent with a properly defined MIB. See SNMP.

modem *Modulator/demodulator*. A device that takes digital data from a computer and encodes it in analog form for transmission over a phone line. See NTU.

modulus The number of unique values available for use as sequence numbers in X.25 packets. For example, if an X.25 packet has a 1 byte control field with 3 bits for each sequence number, the valid range for sequence numbers is 23-1, or 0 to 7, and the modulus is 8.

MTU *Maximum Transmission Unit*. The largest possible unit of data that can be sent on a given physical medium. For local area networks (e.g. Ethernet), the MTU is determined by the network hardware. For wide area networks using serial lines, the MTU is determined by software. The MTU of Ethernet is 1500 bytes. See fragmentation.

multi-homed gateway A dual homed gateway is a system that has two or more network interfaces, each of which is connected to a different network. In firewall configurations, a multi homed gateway usually acts to block or filter some or all of the traffic trying to pass between the networks.

multicast A special form of broadcast where copies of the packet are delivered to only a subset of all possible destinations. See broadcast, directed broadcast, unicast.



multidrop A method of communication where more than two devices may be simultaneously connected to one serial link.

N

NAK *Negative acknowledgement.* A response sent to indicate unsuccessful reception of information. Usually, a NAK triggers retransmission of the lost data. See ACK.

name resolution The process of mapping a name into the corresponding address. See DNS.

NBMA *Non-broadcast Multi-access.* A network topology with multiple access points, such as X.25, that does not support broadcasting, or in which broadcasting is not feasible.

NCP *Network Control Protocol.* A protocol forming part of the Point-to-Point Protocol, used to establish and configure different network layer protocols running over point-to-point links. Each network layer protocol (e.g. IP) has its own associated NCP.

network A computer network is a data communications system which interconnects computer systems at various different sites. A network may be composed of any combination of LANs or WANs.

network address The network portion of an IP address. For a class A network, the network address is the first byte of the IP address. For a class B network, the network address is the first two bytes of the IP address. For a class C network, the network address is the first three bytes of the IP address. In each case, the remainder is the host address. In the Internet, assigned network addresses are globally unique. See IP address.

network number See network address.

network-level firewall A firewall in which traffic is examined at the network protocol packet level.

NIC *Network Information Center.* A group at SRI International, Menlo Park, CA, responsible for providing users with information about TCP/IP and the connected Internet. The machine named NIC.DDN.MIL is an online archive of RFCs and other documents related to TCP/IP.

node An addressable device attached to a computer network. See host, router.

NTU *Network Terminating Unit.* A device that takes digital data from a computer and encodes it for transmission over digital telecommunication lines. It is the equivalent of a modem for modern digital links. See modem.

O

octet An octet is 8 bits. This term is used in networking, rather than byte, because some systems have bytes that are not 8 bits long.



P

packet The unit of data sent across a network. “Packet” is a generic term used to describe units of data at all levels of the protocol stack, but it is most correctly used to describe application data units. See datagram, frame.

packet switching A communications paradigm in which packets (messages) are individually routed between hosts, with no previously established communication path.

PAD *Packet Assembler Disassembler*. A term used with X.25 networks to refer to a terminal multiplexer device that forms a connection between terminals and hosts across an X.25 network. A PAD accepts characters from a terminal and sends them across an X.25 network in packets, and it accepts packets from an X.25 network, extracts the characters, and sends them to a terminal.

parity A method of checking the integrity of characters transmitted serially. It does this by defining an extra bit whose value is set to ensure either an even (even parity) or odd (odd parity) number of ‘1’ bits in the character.

patch A piece of computer code used to correct or enhance an existing piece of code. In the router, patches are applied by “overlying” them on existing code in RAM. The patches are loaded into the router using a process called *downline loading*.

PDU *Protocol Data Unit*. A packet containing a protocol-specific header followed by user data.

perimeter-based security The technique of securing a network by controlling access to all entry and exit points of the network.

ping *Packet InterNet Groper*. A program used to test reachability of destinations by sending them an ICMP echo request and waiting for a reply. The term is used as a verb: “Ping host X to see if it is up!”.

Point-to-Point Protocol See PPP.

PPP *Point-to-Point Protocol*. PPP provides a method for transmitting packets over serial point-to-point links.

policy Organization-level rules governing acceptable use of computing resources, security practices, and operational procedures.

port The abstraction used by Internet transport protocols to distinguish among multiple simultaneous connections to a single destination host. A port is a transport layer demultiplexing value. Each application has a unique port number associated with it. It is also used to refer to one of the physical network connectors on the router.

privacy The property of ensuring that traffic on a secured connection may not be read by unauthorised persons. Privacy is normally ensured using cryptographic techniques (encryption), such as DES, Triple-DES, etc.

privilege A term used in computing to refer the access rights or level of trusted afforded to a user of the computer system. A privileged user has access to “more powerful” commands which may (adversely) affect the operation of the system or the activities of other users. The router has two levels of privilege,



MANAGER and USER. Users with USER privilege (most users) have access to a limited subset of the commands available to MANAGER level users.

prompt A text string displayed on a terminal by a computer to indicate that it is ready to receive the next command from the user.

protocol A formal description of message formats and the rules two computers must follow to exchange those messages. Protocols can describe low-level details of machine-to-machine interfaces (e.g., the order in which bits and bytes are sent across a wire) or high-level exchanges between allocation programs (e.g., the way in which two programs transfer a file across the Internet).

protocol stack A layered set of protocols which work together to provide a set of network functions. See layer, protocol.

proxy A software agent that acts on behalf of a user. Typical proxies accept a connection from a user, make a decision as to whether or not the user is permitted to use the proxy, performs any additional authentication, and then completes a connection on behalf of the user to a remote destination.

proxy ARP The technique in which one machine, usually a router, answers ARP requests intended for another machine. By “faking” its identity, the router accepts responsibility for routing packets to the “real” destination. Proxy ARP allows a site to use a single IP address with two physical networks. Subnetting would normally be a better solution.

PSN *Packet Switch Node.* A dedicated computer whose purpose is to accept, route and forward packets in a packet switched network. See packet switching.

PVC *Permanent Virtual Circuit.* An X.25 virtual circuit that is permanently established.

Q

QOS *Quality of Service.* A measure of the quality of a transmission system, in terms of reliability and availability.

queue A list of packets awaiting processing.

R

RARP *Reverse Address Resolution Protocol.* An IP protocol which provides the reverse function of ARP. RARP maps a hardware (MAC) address to an internet address. It is used primarily by diskless nodes when they boot to find their internet address. See ARP, IP address, MAC address.

reassembly The process in which a previously fragmented packet is reassembled before being passed to the transport layer. See fragmentation.

repeater A device which propagates electrical signals from one cable to another without making routing decisions or providing packet filtering. See bridge and router.

RFC *Request For Comments.* The document series, begun in 1969, which describes the Internet suite of protocols and related experiments. Not all RFCs describe Internet standards, but all Internet standards are written up as RFCs.



RIP *Routing Information Protocol*. A distance vector, as opposed to link state, routing protocol. It is an Internet standard interior gateway protocol (IGP).

route The path that network traffic takes from the source to the destination. It may include many gateways, routers, hosts and physical networks.

route table A table listing information about routes to other hosts or networks, such as the remote network or host address, the interface down which the route exists, the distance to the remote address and the cost of sending data over the route.

router A system responsible for making decisions about which of several paths network (or Internet) traffic will follow. To do this it uses a routing protocol to gain information about the network, and algorithms to choose the best route based on several criteria known as "routing metrics." See gateway, bridge and repeater.

RS-232 An EIA (Electronics Industry Association) standard that specifies the electrical characteristics of low speed interconnections between terminals and computers or between two computers.

S

SAP *Service Access Point*. The point at which the services of an network layer are made available to the next higher layer.

screened host A host on a network behind a screening router. The degree to which a screened host may be accessed depends on the screening rules in the router.

screened subnet A subnet behind a screening router. The degree to which the subnet may be accessed depends on the screening rules in the router.

screening router A router configured to permit or deny traffic based on a set of permission rules installed by the administrator.

serial A method of transmission in which each bit of information is sent sequentially on a single channel rather than simultaneously as in parallel transmission.

server A network device that provides services to client stations. Examples include file servers and print servers.

service A term used with the router to refer to a connection to another port on (another) router, used to access dialup modems, hosts that do not support TCP/IP and other asynchronous devices.

service table A table listing information about services available on the network, such as the service's address.

session stealing See IP splicing.

SNMP *Simple Network Management Protocol*. The Internet standard protocol developed to manage nodes on an IP network. See MIB.

social engineering An attack based on deceiving users or administrators at the target site. Social engineering attacks are typically carried out by telephoning



users or operators and pretending to be an authorized user, to attempt to gain illicit access to systems.

socket An entry point to a program. User programs communicate with transport providers such as UDP and TCP by means of sockets. Each user typically has a separate socket.

source quench A congestion control technique in which a machine experiencing congestion sends a message back to the source of the packets causing the congestion requesting that the source stop transmitting.

spoofing Impersonating a host by sending fake traffic claiming to be from its address. Spoofing is used to hijack existing connections and exploit trust relationships between hosts.

stop bits A technique used in asynchronous serial communications in which 1, 1.5 or 2 bits are transmitted after the start bit, a variable number of data bits and optional parity bit are transmitted. It is designed to frame the character.

subnet A portion of a network, which may be a physically independent network segment, which shares a network address with other portions of the network and is distinguished by a subnet number. A subnet is to a network what a network is to an internet.

subnet address The subnet portion of an IP address. In a subnetted network, the host portion of an IP address is split into a subnet portion and a host portion using an address or subnet mask. See subnet mask, IP address, network address.

subnet mask A bit mask used to select bits from an Internet address for subnet addressing. The mask is 32 bits long and selects the network portion of the Internet address and one or more bits of the local portion. Sometimes called address mask.

SVC *Switched Virtual Circuit*. A virtual circuit that is set up on demand.

T

TA *Terminal Adaptor*. A device enabling the connection of a router without a native ISDN interface to an ISDN circuit.

TCP *Transmission Control Protocol*. The TCP/IP standard transport layer protocol in the Internet suite of protocols, providing reliable, connection-oriented, full-duplex streams. It uses IP for delivery.

TCP/IP Protocol Suite *Transmission Control Protocol over Internet Protocol*. This is a common shorthand which refers to the suite of transport and application protocols which runs over IP. See IP, ICMP, TCP, UDP, FTP, Telnet, SNMP.

Telnet The virtual terminal protocol in the TCP/IP suite of protocols, which allows users of one host to log into a remote host and interact as normal terminal users of that host.

terminal server A device which connects many terminals to a LAN through one network connection. A terminal server can also connect many network users to its asynchronous ports for dial-out capabilities and printer access.



terminator A device placed on a length of coaxial cable to ensure electrical reflections from an un-terminated end are reduced. An Ethernet cable must have exactly two (50Ω) terminators — one at each end of the cable.

TFTP *Trivial File Transfer Protocol*. The TCP/IP standard protocol for file transfer with minimal capability and minimum overhead, based on UDP. It is often used by diskless workstations that keep software in ROM and use it to bootstrap themselves. It is used in the router for downloading patches.

time to live See TTL.

topology A network topology shows the computers and the links between them. A network layer must know the current network topology to be able to route packets to their final destination.

TOS *Type Of Service routing*. A routing scheme in which the choice of route depends on the characteristics of the underlying network topology as well as the shortest path to the destination.

trojan horse A software entity that appears to do something normal but which in fact contains a trapdoor or attack program.

trusted router A concept used to refer to routers from which the router will accept routing information.

TTL *Time To Live*. A technique used in best-effort delivery systems to avoid endlessly looping packets. For example, each IP datagram has a field in the header which indicates how long this packet should be allowed to survive before being discarded. It is primarily used as a hop count. Each time a router processes the packet, it decrements the time to live value. When the value reaches zero, the router discards the packet.

tunnelling A method of transporting a diverse collection of protocols inside a single logical connection.

tunnelling router A router or system capable of routing traffic by encrypting it and encapsulating it for transmission across an untrusted network, for eventual de-encapsulation and decryption.

U

UDP *User Datagram Protocol*. A transport layer protocol in the TCP/IP suite of protocols. UDP, like TCP, uses IP for delivery; however, unlike TCP, UDP provides for exchange of datagrams without acknowledgements or guaranteed delivery.

unicast A packet broadcast to a single host attached to the network. See broadcast, directed broadcast, multicast.

URL *Uniform Resource Locator*. A standard format for specifying the name, type and location of documents and resources on an internet. The syntax is `type://host.domain[:port]/path/filename`, where `type` specifies the type of document or resource (e.g. `http` is a file on a WWW server; `file` is a file on an anonymous FTP server; `telnet` is a connection to a Telnet-based service). See WWW.



V

virtual network perimeter A network that appears to be a single protected network behind firewalls, but which actually encompasses encrypted virtual links over untrusted networks.

virus A self-replicating code segment. Viruses may or may not contain attack programs or trapdoors.

VT-100 A popular model of DEC terminal. Many third party vendors make VT-100 compatible terminals. The term VT-100 is also used to describe the characteristics of terminals that may be connected to a device.

W

WAN *Wide Area Network*. Any physical network technology that spans large geographic distances. WANs usually operate a slower speeds than LANs. See LAN.

well-known port Any of a set of protocol port numbers preassigned for specific uses by transport level protocols, such as TCP and UDP. Examples of well-known port numbers include Telnet (23) and LPD (515).

window In general, a term used to describe a type of flow control mechanism in a network protocol. In an X.25 network, it is the number of unacknowledged packets that may be sent by a DTE, and is less than the modulus for the network. In a TCP/IP network, it is the number of octets that a station is prepared to receive.

wiretapping A generic name given to methods of electronic eavesdropping on traffic as it traverses a network. Wiretapping can compromise reusable passwords and reveal other sensitive information from a login session.

WWW *World Wide Web*. A hypertext-based, distributed information system based on a client-server architecture. Web browsers (client applications) request documents from Web servers. Documents may contain text, graphics and audiovisual data, as well as links to other documents and services. Web servers and documents are identified by URLs (Uniform Resource Locators). See URL.

X

X.25 The CCITT standard protocol for transport level network service. It provides a reliable stream transmission service.

XON/XOFF A method of flow control using the XON and XOFF characters.

Index

Numerics

802.2, Ethernet encapsulation 2-4

A

- access mode, MIB object 16-5
- ACTIVATE FLASH COMPACTION command 1-26
- ACTIVATE ISDN CALL command 4-42
- ACTIVATE MIOX CIRCUIT command 5-9
- ACTIVATE PPP command 3-27
- ACTIVATE Q931 ASPID command 4-43
- ACTIVATE SCRIPT command 13-5
- ACTIVATE TRIGGER command 10-5
- ADD ALIAS command 1-28
- ADD BOOTP RELAY command 6-43
- ADD DHCP POLICY command 15-4
- ADD DHCP RANGE command 15-9
- ADD FIREWALL POLICY INTERFACE command 17-11
- ADD FIREWALL POLICY NAT command 17-12
- ADD FIREWALL POLICY RULE command 17-14
- ADD IP ARP command 6-44
- ADD IP FILTER command 6-45
- ADD IP HELPER command 6-51
- ADD IP HOST command 6-52
- ADD IP INTERFACE command 6-53
- ADD IP RIP command 6-56
- ADD IP ROUTE command 6-57
- ADD IP ROUTE FILTER command 6-59
- ADD IP ROUTE TEMPLATE command 6-60
- ADD IP TRUSTED command 6-61
- ADD ISDN CALL command 4-44
- ADD ISDN CLILIST command 4-50
- ADD ISDN DOMAINNAME command 4-50
- ADD LAPD command 4-82
- ADD LAPD TEI command 4-51
- ADD LAPD XSPID command 4-51
- ADD LAPD XTEI command 4-52
- ADD LOG OUTPUT command 12-12
- ADD LOG RECEIVE command 12-15
- ADD MIOX CIRCUIT command 5-10, 5-19
- ADD PPP command 3-28
- ADD SCRIPT command 13-2, 13-6
- ADD SNMP COMMUNITY command 16-13
- ADD TDM command 11-3
- ADD TRIGGER command 10-6
- ADD USER command 1-28
- ADD X25T CPAR command 5-11
- adding
 - access lists, firewall 17-14
 - aliases 1-28
 - domain name 4-27, 4-50
 - firewall rules 17-4, 17-14
 - interfaces to firewall 17-4, 17-11
 - NAT translations, firewall 17-6, 17-12
 - physical links to PPP interfaces 3-28
 - route templates 6-60
- address pools, IP 6-32
 - creating 6-62
 - destroying 6-71
 - displaying 6-124
- Address Resolution Protocol (ARP) 6-9
 - adding entries 6-10, 6-44
 - changing entries 6-86
 - deleting entries 6-63
 - displaying entries 6-10, 6-35, 6-108
 - modifying entries 6-86
 - proxy 6-10
 - setting entries 6-86
- address translation group, MIB-II MIB C-11
- addresses
 - Ethernet multicast 2-6, 2-29
 - Internet Protocol (IP) 6-4
 - address pools 6-32, 6-62, 6-71, 6-124
 - classes 6-6
 - format 6-6
 - source 6-4
 - subnet 6-8
 - multicast 2-6, 2-29
 - subnet 6-8
 - X.25 DTE 5-4
- addressing, LAPD 4-13
- administrator name
 - displaying 1-78
 - setting 1-54
- aggregation, PPP links 3-22
- aliases 1-5
 - adding 1-28
 - deleting 1-31



- displaying 1-58
- alliedTelesyn* MIB object C-3
- Always On/Dynamic ISDN (AODI) 4-28 to 4-31
 - components 4-28
 - configuration example 4-29
- AODI. See Always On/Dynamic ISDN (AODI)
- application gateway 17-2
- arBoardIndex* MIB object C-7
- arBoardMaxIndex* MIB object C-7
- arBoardTable* MIB object C-7 to C-9
- arInterfaceBoardIndex* MIB object C-7
- arInterfacePosition* MIB object C-7
- arInterfaces* Group, Allied Telesyn Enterprise MIB C-6
- arInterfaces* MIB object C-4
- arInterfaceTable* MIB object C-7
- arSlotBoardIndex* MIB object C-7
- arSlotSlotIndex* MIB object C-7
- arSlotTable* MIB object C-7
- asynchronous
 - ports
 - configuration 2-7, 2-14 to 2-15, 2-17 to 2-19
 - default port characteristics 2-8
 - disabling ports 2-14
 - enabling ports 2-15
 - purging configuration 2-15
 - resetting counters 2-17
 - resetting history 2-18
 - resetting ports 2-17
 - setting port characteristics 2-7, 2-19
- asynchronous port security 1-11
- asynchronous ports 2-7
 - displaying configuration 2-33
 - encapsulation 2-7
- atfIndex* MIB object C-12
- atNetAddress* MIB object C-12
- atPhysAddress* MIB object C-12
- atRouter* MIB object C-4
- atRouter* sub-tree, Allied Telesyn Enterprise MIB C-4
- authentication
 - SNMP 16-10
 - using RADIUS servers 1-7
 - using TACACS servers 1-7
 - using User Authentication Database 1-7 to 1-8, 1-28, 1-33, 1-35, 1-41, 1-44 to 1-46, 1-48, 1-57, 1-80
- Authentication Protocol Option, LCP 3-5
- authentication protocols, PPP 3-12 to 3-15
- auto-bauding 2-11
- autonomous systems
 - setting AS number 6-87

B

- BACP. See Bandwidth Allocation Control Protocol
- Bandwidth Allocation Control Protocol (BACP) 3-7
- Bandwidth Allocation Protocol (BAP) 3-7
- bandwidth on demand, PPP 3-8, 3-25
- BAP. See Bandwidth Allocation Protocol
- Basic Rate Access
 - configuring 4-7
- Basic Rate Access, ISDN 4-3 to 4-6
- boards* MIB object C-5

- boot script 1-6
- BOOT.CFG 13-2
- booting the router 1-13, 1-49
- BOOTP relay agent 6-29 to 6-30
- BRI Group, Allied Telesyn Enterprise MIB C-8
- bri* MIB object C-8
- BRI physical layer 4-7
- briChanTable* MIB object C-8
- bridgeRouter* MIB object C-3 to C-4
- brIntTable* MIB object C-8
- broadcast forwarding 6-26 to 6-29
 - to a broadcast address 6-28
 - to a unicast address 6-27
- brouterMib* MIB object C-4

C

- call control, ISDN 4-22
- call logging, ISDN 4-27
- CAPI. See Common Application Programmer's Interface (CAPI)
- cc* MIB object C-8
- ccActiveCallTable* MIB object C-8
- ccAttachmentTable* MIB object C-8
- ccBchannelTable* MIB object C-8
- ccCallLogTable* MIB object C-8
- ccCliListTable* MIB object C-8
- ccDetailsTable* MIB object C-8
- centreCOM-AR300* MIB object C-4
- centreCOM-AR300L* MIB object C-4
- centreCOM-AR300LU* MIB object C-4
- centreCOM-AR300U* MIB object C-4
- centreCOM-AR310* MIB object C-4
- centreCOM-AR310U* MIB object C-4
- centreCOM-AR330* MIB object C-4
- centreCOM-AR350* MIB object C-4
- centreCOM-AR370* MIB object C-4
- centreCOM-AR370U* MIB object C-4
- centreCOM-AR390* MIB object C-4
- centreCOM-AR395* MIB object C-4
- centreCOM-AR720* MIB object C-4
- Challenge-Handshake Authentication Protocol (CHAP) 3-13
- CHAP. See Challenge-Handshake Authentication Protocol (CHAP)
- chip_68020_cpu* MIB object C-6
- chip_68302_cpu* MIB object C-6
- chip_68340_cpu* MIB object C-6
- chip_68360_cpu* MIB object C-6
- chip_860T_cpu* MIB object C-6
- chip_flash_1mb* MIB object C-6
- chip_flash_2mb* MIB object C-6
- chip_flash_3mb* MIB object C-6
- chip_flash_4mb* MIB object C-6
- chip_flash_6mb* MIB object C-6
- chip_flash_8mb* MIB object C-6
- chip_pem* MIB object C-6
- chip_ram_12mb* MIB object C-6
- chip_ram_16mb* MIB object C-6
- chip_ram_1mb* MIB object C-6
- chip_ram_20mb* MIB object C-6



- chip_ram_2mb* MIB object C-6
- chip_ram_32mb* MIB object C-6
- chip_ram_3mb* MIB object C-6
- chip_ram_4mb* MIB object C-6
- chip_ram_6mb* MIB object C-6
- chip_ram_8mb* MIB object C-6
- chip_rtc1* MIB object C-6
- chip_rtc2* MIB object C-6
- chip_rtc3* MIB object C-6
- chip_rtc4* MIB object C-6
- chips* MIB object C-6
- CLEAR FLASH TOTALLY command 1-29
- clock
 - displaying 1-80
 - setting 1-56
- CMOT group, MIB-II MIB C-11
- combined notation 16-4
- command
 - aliases 1-5
 - editing 1-4
 - history 1-4
 - processor 1-3
 - prompt 1-4
- command line
 - editing 7-4
 - history 7-4
 - recall 7-4
- commands 1-75
 - ACTIVATE FLASH COMPACTION 1-26
 - ACTIVATE ISDN CALL 4-42, 4-112
 - ACTIVATE MIOX CIRCUIT 5-9
 - ACTIVATE PPP 3-27
 - ACTIVATE Q931 ASPID 4-43
 - ACTIVATE SCRIPT 13-5
 - ACTIVATE TRIGGER 10-5
 - ADD ALIAS 1-28
 - ADD BOOTP RELAY 6-43
 - ADD DHCP POLICY 15-4
 - ADD DHCP RANGE 15-9
 - ADD FIREWALL POLICY INTERFACE 17-11
 - ADD FIREWALL POLICY NAT 17-12
 - ADD FIREWALL POLICY RULE 17-14
 - ADD IP ARP 6-44
 - ADD IP FILTER 6-45
 - ADD IP HELPER 6-51, 6-64
 - ADD IP HOST 6-52
 - ADD IP INTERFACE 6-53
 - ADD IP RIP 6-56
 - ADD IP ROUTE 6-57
 - ADD IP ROUTE FILTER 6-59
 - ADD IP ROUTE TEMPLATE 6-60
 - ADD IP TRUSTED 6-61
 - ADD ISDN CALL 4-44
 - ADD ISDN CLILIST 4-50
 - ADD ISDN DOMAINNAME 4-50
 - ADD LAPD 4-82
 - ADD LAPD TEI 4-51
 - ADD LAPD XSPID 4-51
 - ADD LAPD XTEI 4-52
 - ADD LOG OUTPUT 12-12
 - ADD LOG RECEIVE 12-15
 - ADD MIOX CIRCUIT 5-10
 - ADD PPP 3-28
 - ADD SCRIPT 13-2, 13-6
 - ADD SNMP COMMUNITY 16-13
 - ADD TDM 11-3
 - ADD TRIGGER 10-6
 - ADD USER 1-28
 - ADD X25T CPAR 5-11
 - CLEAR FLASH TOTALLY 1-29
 - CREATE CONFIG 1-30
 - CREATE DHCP POLICY 15-9
 - CREATE DHCP RANGE 15-10
 - CREATE FFILE 1-31
 - CREATE FIREWALL POLICY 17-16
 - CREATE IP POOL 6-62
 - CREATE LOG OUTPUT 12-16
 - CREATE PBX EXTENSION 14-13
 - CREATE PBX GROUP 14-16
 - CREATE PPP 3-31
 - CREATE PPP TEMPLATE 3-36
 - CREATE SNMP COMMUNITY 16-14
 - CREATE TDM 11-4
 - CREATE TRIGGER 10-7
 - CREATE X25T 5-12
 - DEACTIVATE ISDN CALL 4-52
 - DEACTIVATE MIOX CIRCUIT 5-14
 - DEACTIVATE SCRIPT 13-7
 - DELETE ALIAS 1-31
 - DELETE BOOTP RELAY 6-63
 - DELETE DHCP POLICY 15-11
 - DELETE DHCP RANGE 15-15
 - DELETE FFILE 1-32
 - DELETE FILE 1-32
 - DELETE FIREWALL POLICY INTERFACE 17-17
 - DELETE FIREWALL POLICY NAT 17-18
 - DELETE FIREWALL POLICY RULE 17-19
 - DELETE FIREWALL SESSION 17-19
 - DELETE INSTALL 1-33
 - DELETE IP ARP 6-63
 - DELETE IP FILTER 6-64
 - DELETE IP HOST 6-65
 - DELETE IP INTERFACE 6-66
 - DELETE IP RIP 6-67
 - DELETE IP ROUTE 6-68
 - DELETE IP ROUTE FILTER 6-69
 - DELETE IP ROUTE TEMPLATE 6-69
 - DELETE IP TRUSTED 6-69
 - DELETE ISDN CALL 4-53
 - DELETE ISDN CLILIST 4-53
 - DELETE ISDN DOMAINNAME 4-54
 - DELETE LAPD TEI 4-54
 - DELETE LAPD XSPID 4-55
 - DELETE LAPD XTEI 4-55
 - DELETE LOG OUTPUT 12-19
 - DELETE LOG RECEIVE 12-20
 - DELETE MIOX CIRCUIT 5-15
 - DELETE PPP 3-41
 - DELETE SCRIPT 13-8
 - DELETE SNMP COMMUNITY 16-15

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

DELETE TCP 6-70
 DELETE TDM 11-5
 DELETE TRIGGER 10-11
 DELETE USER 1-33
 DELETE X25T CPAR 5-15
 DESTROY DHCP POLICY 15-15
 DESTROY DHCP RANGE 15-16
 DESTROY FIREWALL POLICY 17-20
 DESTROY IP POOL 6-71
 DESTROY LOG OUTPUT 12-20
 DESTROY PBX EXTENSION 14-17
 DESTROY PBX GROUP 14-18
 DESTROY PPP 3-41
 DESTROY PPP TEMPLATE 3-42
 DESTROY SNMP COMMUNITY 16-16
 DESTROY TDM 11-5
 DESTROY TRIGGER 10-11
 DESTROY X25T 5-16
 DISABLE BOOTP RELAY 6-71
 DISABLE BRI CTEST 4-56
 DISABLE BRI DEBUG 4-56
 DISABLE BRI TEST 4-57
 DISABLE DHCP 15-16
 DISABLE ENCO COMPSTATISTICS 8-5
 DISABLE ENCO DEBUGGING 8-6
 DISABLE FIREWALL 17-20
 DISABLE FIREWALL NOTIFY 17-21
 DISABLE FIREWALL POLICY 17-21
 DISABLE HTTP SERVER 1-34
 DISABLE INTERFACE LINKTRAP 2-13
 DISABLE IP 6-72
 DISABLE IP DEBUG 6-72
 DISABLE IP DNSRELAY 6-73
 DISABLE IP ECHOREPLY 6-73
 DISABLE IP FOFILTER 6-73
 DISABLE IP FORWARDING 6-74
 DISABLE IP HELPER 6-74
 DISABLE IP INTERFACE 6-75
 DISABLE IP REMOTEASSIGN 6-75
 DISABLE IP ROUTE 6-76
 DISABLE IP SRCROUTE 6-76
 DISABLE ISDN CALL 4-58
 DISABLE ISDN LOG 4-58
 DISABLE LOG 12-21
 DISABLE LOG GENERATION 12-21
 DISABLE LOG OUTPUT 12-21
 DISABLE LOG RECEPTION 12-22
 DISABLE MIOX CIRCUIT 5-16
 DISABLE PBX DEBUG 14-18
 DISABLE PORT 2-14
 DISABLE PPP 3-42
 DISABLE PPP DEBUG 3-43
 DISABLE PPP TEMPLATE DEBUG 3-43
 DISABLE Q931 DEBUG 4-59
 DISABLE RAPI 4-60
 DISABLE SNMP 16-16
 DISABLE SNMP AUTHENTICATE_TRAP 16-16
 DISABLE SNMP COMMUNITY 16-17
 DISABLE TEST INTERFACE 9-8
 DISABLE TRIGGER 10-12
 DISABLE USER 1-35
 DUMP 1-35
 EDIT 1-37, 13-3
 ENABLE BOOTP RELAY 6-77
 ENABLE BRI CTEST 4-60
 ENABLE BRI DEBUG 4-61
 ENABLE BRI TEST 4-62
 ENABLE DHCP 15-17
 ENABLE ENCO COMPSTATISTICS 8-6
 ENABLE ENCO DEBUGGING 8-7
 ENABLE FIREWALL 17-22
 ENABLE FIREWALL POLICY 17-22
 ENABLE HTTP DEBUG 1-40
 ENABLE HTTP SERVER 1-40
 ENABLE INTERFACE LINKTRAP 2-14
 ENABLE IP 6-77
 ENABLE IP DEBUG 6-78
 ENABLE IP DNSRELAY 6-78
 ENABLE IP ECHOREPLY 6-78
 ENABLE IP FOFILTER 6-78
 ENABLE IP FORWARDING 6-79
 ENABLE IP HELPER 6-80
 ENABLE IP INTERFACE 6-80
 ENABLE IP REMOTEASSIGN 6-81
 ENABLE IP ROUTE 6-81
 ENABLE IP SRCROUTE 6-82
 ENABLE ISDN CALL 4-65
 ENABLE ISDN LOG 4-65
 ENABLE LOG 12-22
 ENABLE LOG GENERATION 12-23
 ENABLE LOG OUTPUT 12-23
 ENABLE LOG RECEPTION 12-23
 ENABLE MIOX CIRCUIT 5-17
 ENABLE PBX DEBUG 14-19
 ENABLE PORT 2-15
 ENABLE PPP 3-44
 ENABLE PPP DEBUG 3-45
 ENABLE PPP TEMPLATE DEBUG 3-46
 ENABLE Q931 ASPID 4-66
 ENABLE Q931 DEBUG 4-66
 ENABLE RAPI 4-71
 ENABLE SNMP 16-17
 ENABLE SNMP AUTHENTICATE_TRAP 16-18
 ENABLE SNMP COMMUNITY 16-18
 ENABLE TEST INTERFACE 9-9
 ENABLE TRIGGER 10-12
 ENABLE USER 1-41
 FLUSH LOG OUTPUT 12-24
 HELP 1-41
 Internet Protocol (IP) 6-43 to 6-147, 16-12
 LOAD 1-42, 13-3
 LOGIN 1-44
 LOGOFF 1-45
 MODIFY 1-45
 PING 6-82
 Point-to-Point Protocol (PPP) 3-27
 PURGE BOOTP RELAY 6-84
 PURGE IP 6-84
 PURGE LOG 12-24
 PURGE PORT 2-15



PURGE PPP 3-47
 PURGE TDM 11-6
 PURGE TRIGGER 10-13
 PURGE USER 1-46
 RENAME 1-46
 RESET BRI 4-72
 RESET BRI COUNTERS 4-72
 RESET ENCO COUNTERS 8-7
 RESET ETH 2-16
 RESET ETH COUNTERS 2-16
 RESET HTTP SERVER 1-47
 RESET IP 6-84
 RESET IP COUNTER 6-85
 RESET IP INTERFACE 6-85
 RESET LOADER 1-47
 RESET PORT 2-17
 RESET PORT COUNTERS 2-17
 RESET PORT HISTORY 2-18
 RESET PPP 3-48
 RESET Q931 4-73
 RESET TEST INTERFACE 9-10
 RESET USER 1-48
 RESET X25T 5-18
 RESTART 1-49
 SET BOOTP MAXHOPS 6-86
 SET BRI 4-73
 SET CONFIG 1-49
 SET DHCP POLICY 15-17
 SET ENCO SW 8-7
 SET FIREWALL POLICY RULE 17-24
 SET HELP 1-50
 SET INSTALL 1-50
 SET INTERFACE TRAPLIMIT 2-18
 SET IP ARP 6-86
 SET IP AUTONOMOUS 6-87
 SET IP FILTER 6-88
 SET IP HOST 6-91
 SET IP INTERFACE 6-92
 SET IP LOCAL 6-94
 SET IP NAMESERVER 6-95
 SET IP RIP 6-96
 SET IP RIPTIMER 6-97
 SET IP ROUTE 6-98
 SET IP ROUTE FILTER 6-100
 SET IP ROUTE TEMPLATE 6-101
 SET IP SECONDARYNAMESERVER 6-102
 SET ISDN CALL 4-75
 SET ISDN DOMAINNAME 4-80
 SET ISDN LOG 4-81
 SET LOADER 1-51
 SET LOG OUTPUT 12-25
 SET LOG RECEIVE 12-29
 SET LOG UTCOFFSET 12-30
 SET MANAGER PORT 1-53
 SET MIOX 5-18
 SET MIOX CIRCUIT 5-19
 SET PASSWORD 1-54
 SET PBX 14-20
 SET PBX EXTENSION 14-21
 SET PBX GROUP 14-24
 SET PING 6-103
 SET PORT 2-19
 SET PPP 3-49
 SET PPP TEMPLATE 3-54
 SET Q931 4-84
 SET SCRIPT 13-2, 13-10
 SET SNMP COMMUNITY 16-19
 SET SYSTEM CONTACT 1-54
 SET SYSTEM LOCATION 1-55
 SET SYSTEM NAME 1-55
 SET SYSTEM TERRITORY 1-56, 4-16, 4-33, 4-35
 SET TELNET 7-6
 SET TIME 1-56
 SET TRACE 6-104
 SET TRIGGER 10-13
 SET TTY 7-7
 SET USER 1-57
 SET X25T 5-21
 SET X25T CPAR 5-22
 SHOW ALIAS 1-58
 SHOW BOOTP RELAY 6-105
 SHOW BRI CONFIGURATION 4-86
 SHOW BRI COUNTERS 4-88
 SHOW BRI CTEST 4-94
 SHOW BRI DEBUG 4-95
 SHOW BRI STATE 4-96
 SHOW BRI TEST 4-100
 SHOW BUFFER 1-59
 SHOW CONFIG 1-61
 SHOW CPU 1-63
 SHOW DEBUG 1-63
 SHOW DHCP 15-22
 SHOW DHCP CLIENT 15-23
 SHOW DHCP POLICY 15-24
 SHOW DHCP RANGE 15-25
 SHOW ENCO 8-8
 SHOW ENCO CHANNEL 8-9
 SHOW ENCO COUNTERS 8-13
 SHOW ETH CONFIGURATION 2-22
 SHOW ETH COUNTERS 2-23
 SHOW ETH MACADDRESS 2-29
 SHOW ETH RECEIVE 2-29
 SHOW EXCEPTION 1-64
 SHOW FFIL 1-65, 1-67
 SHOW FIREWALL 17-25
 SHOW FIREWALL POLICY 17-26
 SHOW FIREWALL POLICY SESSION 17-32
 SHOW FLASH 1-67
 SHOW FLASH PHYSICAL 1-69
 SHOW HTTP CLIENT 1-69
 SHOW HTTP DEBUG 1-70
 SHOW HTTP SERVER 1-72
 SHOW HTTP SESSION 1-71
 SHOW INSTALL 1-74
 SHOW INTERFACE 2-30
 SHOW IP 6-106
 SHOW IP ARP 6-108
 SHOW IP COUNTER 6-109
 SHOW IP DEBUG 6-116
 SHOW IP FILTER 6-117

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

SHOW IP HELPER 6-119
 SHOW IP HOST 6-120
 SHOW IP INTERFACE 6-121
 SHOW IP POOL 6-124
 SHOW IP RIP 6-126
 SHOW IP RIP COUNTER 6-128
 SHOW IP RIPTIMER 6-127
 SHOW IP ROUTE 6-130
 SHOW IP ROUTE FILTER 6-134
 SHOW IP ROUTE TEMPLATE 6-135
 SHOW IP TRUSTED 6-136
 SHOW IP UDP 6-136
 SHOW ISDN CALL 4-103
 SHOW ISDN CLILIST 4-107
 SHOW ISDN DOMAINNAME 4-108
 SHOW ISDN LOG 4-109
 SHOW LAPD 4-110
 SHOW LAPD COUNT 4-112
 SHOW LAPD STATE 4-114
 SHOW LOG 12-31 to 12-32
 SHOW LOG COUNTERS 12-38
 SHOW LOG OUTPUT 12-40, 12-42
 SHOW LOG QUEUE 12-44, 12-47
 SHOW LOG RECEIVE 12-45
 SHOW LOG STATUS 12-46
 Logging Facility 12-46
 SHOW MANAGER PORT 1-77
 SHOW MIOX 5-23
 SHOW MIOX CIRCUIT 5-26
 SHOW MIOX COUNT 5-24
 SHOW PBX 14-25
 SHOW PBX CALL 14-26
 SHOW PBX EXTENSION 14-27
 SHOW PBX GROUP 14-29
 SHOW PING 6-137
 SHOW PORT 2-33
 SHOW PPP 3-58
 SHOW PPP CONFIG 3-59
 SHOW PPP COUNT 3-65
 SHOW PPP DEBUG 3-75
 SHOW PPP IDLETIMER 3-76
 SHOW PPP LIMITS 3-77
 SHOW PPP MULTILINK 3-78
 SHOW PPP NAMESERVER 3-80
 SHOW PPP TEMPLATE 3-80
 SHOW PPP TXSTATUS 3-84
 SHOW Q931 4-114
 SHOW Q931 SPID 4-117
 SHOW SCRIPT 13-2, 13-11
 SHOW SNMP 16-20
 SHOW SNMP COMMUNITY 16-22
 SHOW STARTUP 1-77
 SHOW SYSTEM 1-78
 SHOW TCP 6-139
 SHOW TDM 11-6
 SHOW TEST 9-10
 SHOW TIME 1-80
 SHOW TRACE 6-143
 SHOW TRIGGER 10-16
 SHOW TTY 7-8

SHOW USER 1-80
 SHOW X25T 5-30
 SHOW X25T CPAR 5-34
 STOP PING 6-145
 STOP TRACE 6-145
 TELNET 7-11
 terminal server 7-6
 TRACE 6-146
 UPLOAD 1-82
 WAIT 13-8, 13-12
 X.25 5-8
 common D channel signalling, ISDN 4-15
 community, SNMP 16-8
 compacting FLASH memory 1-17, 1-26
 compression
 Point-to-Point Protocol (PPP) 3-25
 Van Jacobson's 3-18
 compression. See Compression and Encryption Services;
 Link Compression and Encryption; IP Security (IPsec)
 Compression and Encryption Services 8-1 to 8-21
 See also Link Compression and Encryption; IP Security
 (IPsec)
 See also IP Security (IPsec)
 channels
 ENCO 8-9
 commands
 DISABLE ENCO COMPSTATISTICS 8-5
 DISABLE ENCO DEBUGGING 8-6
 ENABLE ENCO COMPSTATISTICS 8-6
 ENABLE ENCO DEBUGGING 8-7
 RESET ENCO COUNTERS 8-7
 SET ENCO SW 8-7
 SHOW ENCO 8-8
 SHOW ENCO CHANNEL 8-9
 SHOW ENCO COUNTERS 8-13
 compression
 header 8-3
 history 8-2
 Lempel-Ziv algorithm 8-2
 link 8-2
 lossless 8-2
 lossy 8-2
 payload 8-3
 Predictor 8-7
 services 8-4
 software 8-4, 8-7
 speed 8-7
 STAC LZS 8-7
 statistics 8-5 to 8-6
 Van Jacobson 8-3
 counters
 displaying 8-13
 resetting 8-7
 data compression 8-2 to 8-3
 debugging
 disabling ENCO 8-6
 enabling ENCO 8-7
 disabling
 debugging 8-6
 displaying



- ENCO counters 8-13
- displaying counters 8-13
- enabling
 - debugging 8-7
- ENCO channels 8-9
- ENCO services 8-4, 8-8 to 8-9
- header compression 8-3
- history, compression 8-2
- Lempel-Ziv compression algorithm 8-2
- link compression 8-2
- lossless data compression 8-2
- lossy data compression 8-2
- payload compression 8-3
- PPP 8-5
- Predictor compression 8-7
- resetting
 - ENCO counters 8-7
- resetting counters 8-7
- services 8-4, 8-8 to 8-9
 - compression 8-4
- software compression 8-4, 8-7
- speed, software compression 8-7
- STAC LZS compression 8-7
- statistics, compression 8-5 to 8-6
- user modules
 - PPP 8-5
 - X.25 8-5
- Van Jacobson header compression 8-3
- X.25 8-5
- compression statistics 8-5 to 8-6
- configFile* MIB object C-9 to C-10
- configuration
 - asynchronous ports 2-7
 - displaying dynamic 1-61
 - Ethernet 2-16
 - examples
 - Internet Protocol (IP) 6-33
 - Internet Protocol (IP) 6-33
 - examples 6-33
 - retrieving 1-6
 - saving dynamic 1-6, 1-30
 - setting default 1-49
- configuration examples
 - AODI 4-29
 - DHCP 15-3
 - firewall 17-8 to 17-11
 - installing a compressed release 1-25
 - installing a standard release 1-24
 - Logging Facility 12-9
 - patches 1-24
 - Point-to-Point Protocol (PPP) 3-20
 - SNMP 16-11
 - software releases 1-24
 - Time Division Multiplexing (TDM) 11-2
- configuring
 - Basic Rate Access, ISDN 4-7
 - DHCP 15-3
 - ISDN 4-33
- contact name
 - displaying 1-78

- setting 1-54
- control protocols, Point-to-Point Protocol 3-3 to 3-5
- control structures, in scripts 13-4, 13-8
- conventions, file naming 1-15 to 1-16
- counters
 - displaying
 - ENCO 8-13
 - ENCO
 - displaying 8-13
 - resetting 8-7
 - ETH 2-16, 2-23
 - Ethernet 2-16
 - Internet Protocol (IP) 6-109
 - resetting 6-85
 - Point-to-Point Protocol (PPP) 3-19
 - resetting
 - Internet Protocol (IP) 6-85
 - resetting
 - ENCO 8-7
- couters
 - displaying PPP activity 3-17
 - resetting PPP activity 3-17, 3-48
- CPU utilisation 1-13, 1-63
- CREATE CONFIG command 1-30
- CREATE DHCP POLICY command 15-9
- CREATE DHCP RANGE command 15-10
- CREATE FFILE command 1-31
- CREATE FIREWALL POLICY command 17-16
- CREATE IP POOL command 6-62
- CREATE LOG OUTPUT command 12-16
- CREATE PBX EXTENSION command 14-13
- CREATE PBX GROUP command 14-16
- CREATE PPP command 3-31
- CREATE PPP TEMPLATE command 3-36
- CREATE SNMP COMMUNITY command 16-14
- CREATE TDM command 11-4
- CREATE TRIGGER command 10-7
- CREATE X25T command 5-12
- createConfigFile* MIB object C-10
- creating
 - extensions 14-13
 - files 1-16
 - FLASH files 1-31
 - groups, PBX 14-16
 - IP address pools 6-62
 - PPP interfaces 3-31
 - PPP templates 3-36
 - templates 3-36

D

- Data Communication Equipment (DCE) 5-2
- data compression. See Compression and Encryption Services
- data over voice 4-32
- Data Terminal Equipment (DTE) 5-2
- date
 - displaying 1-80
 - setting 1-56
- DEACTIVATE ISDN CALL command 4-52
- DEACTIVATE MIOX CIRCUIT command 5-14



- DEACTIVATE SCRIPT command 13-7
- debugging
 - disabling
 - ENCO 8-6
 - enabling
 - ENCO 8-7
 - ENCO
 - disabling 8-6
 - enabling 8-7
 - firewall 17-6
 - PBX 14-18 to 14-19
 - PPP interfaces 3-43, 3-45
 - PPP links 3-17, 3-75
 - PPP templates 3-43, 3-46
 - router 1-63
 - templates 3-43, 3-46
- debugging, Internet Protocol (IP) 6-24 to 6-25, 6-72, 6-78, 6-116
- default configuration, setting 1-49
- DELETE ALIAS command 1-31
- DELETE BOOTP RELAY command 6-63
- DELETE DHCP POLICY command 15-11
- DELETE DHCP RANGE command 15-15
- DELETE FFILE command 1-32
- DELETE FILE command 1-32
- DELETE FIREWALL POLICY INTERFACE command 17-17
- DELETE FIREWALL POLICY NAT command 17-18
- DELETE FIREWALL POLICY RULE command 17-19
- DELETE FIREWALL SESSION command 17-19
- DELETE INSTALL command 1-33
- DELETE IP ARP command 6-63
- DELETE IP FILTER command 6-64
- DELETE IP HELPER command 6-64
- DELETE IP HOST command 6-65
- DELETE IP INTERFACE command 6-66
- DELETE IP RIP command 6-67
- DELETE IP ROUTE command 6-68
- DELETE IP ROUTE FILTER command 6-69
- DELETE IP ROUTE TEMPLATE command 6-69
- DELETE IP TRUSTED command 6-69
- DELETE ISDN CALL command 4-53
- DELETE ISDN CLILIST command 4-53
- DELETE ISDN DOMAINNAME command 4-54
- DELETE LAPD TEI command 4-54
- DELETE LAPD XSPID command 4-55
- DELETE LAPD XTEI command 4-55
- DELETE LOG OUTPUT command 12-19
- DELETE LOG RECEIVE command 12-20
- DELETE MIOX CIRCUIT command 5-15
- DELETE PPP command 3-41
- DELETE SCRIPT command 13-8
- DELETE SNMP COMMUNITY command 16-15
- DELETE TCP command 6-70
- DELETE TDM command 11-5
- DELETE TRIGGER command 10-11
- DELETE USER command 1-33
- DELETE X25T CPAR command 5-15
- deleting
 - access lists, firewall 17-19
 - aliases 1-31
 - domain name 4-27, 4-54
 - files 1-16, 1-32
 - firewall rules 17-4, 17-19
 - FLASH files 1-32
 - install information 1-33
 - interfaces from firewall 17-4, 17-17
 - NAT translations, firewall 17-6, 17-18
 - physical links from PPP interfaces 3-41
 - PPP interfaces 3-41
 - route templates 6-69
- deprecated status 16-6
- descriptor 16-4
- descriptor, MIB object 16-4
- DESTROY DHCP POLICY command 15-15
- DESTROY DHCP RANGE command 15-16
- DESTROY FIREWALL POLICY command 17-20
- DESTROY IP POOL command 6-71
- DESTROY LOG OUTPUT command 12-20
- DESTROY PBX EXTENSION command 14-17
- DESTROY PBX GROUP command 14-18
- DESTROY PPP command 3-41
- DESTROY PPP TEMPLATE command 3-42
- DESTROY SNMP COMMUNITY command 16-16
- DESTROY TDM command 11-5
- DESTROY TRIGGER command 10-11
- DESTROY X25T command 5-16
- destroying
 - extensions 14-17
 - groups, PBX 14-18
 - IP address pools 6-71
 - PPP interfaces 3-41
 - PPP templates 3-42
 - templates 3-42
- DHCP client 6-11, 6-54, 6-73, 6-78, 6-93
- DHCP. See Dynamic Host Configuration Protocol (DHCP)
- dial-on-demand, PPP 3-8, 3-23
- directory of files 1-16
- DISABLE BOOTP RELAY command 6-71
- DISABLE BRI CTEST command 4-56
- DISABLE BRI DEBUG command 4-56
- DISABLE BRI TEST command 4-57
- DISABLE DHCP command 15-16
- DISABLE ENCO COMPSTATISTICS command 8-5
- DISABLE ENCO DEBUGGING command 8-6
- DISABLE FIREWALL command 17-20
- DISABLE FIREWALL NOTIFY command 17-21
- DISABLE FIREWALL POLICY command 17-21
- DISABLE HTTP SERVER command 1-34
- DISABLE INTERFACE LINKTRAP command 2-13
- DISABLE IP command 6-72
- DISABLE IP DEBUG command 6-72
- DISABLE IP DNSRELAY command 6-73
- DISABLE IP ECHOREPLY command 6-73
- DISABLE IP FOFILTER command 6-73
- DISABLE IP FORWARDING command 6-74
- DISABLE IP HELPER command 6-74
- DISABLE IP INTERFACE command 6-75
- DISABLE IP REMOTEASSIGN command 6-75
- DISABLE IP ROUTE command 6-76
- DISABLE IP SRCROUTE command 6-76



DISABLE ISDN CALL command 4-58
 DISABLE ISDN LOG command 4-58
 DISABLE LOG command 12-21
 DISABLE LOG GENERATION command 12-21
 DISABLE LOG OUTPUT command 12-21
 DISABLE LOG RECEPTION command 12-22
 DISABLE MIOX CIRCUIT command 5-16
 DISABLE PBX DEBUG command 14-18
 DISABLE PORT command 2-14
 DISABLE PPP command 3-42
 DISABLE PPP DEBUG command 3-43
 DISABLE PPP TEMPLATE DEBUG command 3-43
 DISABLE Q931 DEBUG command 4-59
 DISABLE RAPI command 4-60
 DISABLE SNMP AUTHENTICATE_TRAP command 16-16
 DISABLE SNMP command 16-16
 DISABLE SNMP COMMUNITY command 16-17
 DISABLE TEST INTERFACE command 9-8
 DISABLE TRIGGER command 10-12
 DISABLE USER command 1-35

disabling

- debugging
 - ENCO 8-6
 - PBX 14-18
- debugging, PPP interfaces 3-43
- debugging, PPP templates 3-43
- ENCO debugging 8-6
- firewall 17-3, 17-20
- firewall accounting 17-21
- firewall debugging 17-6, 17-21
- hardware tests 9-8
- interfaces, PPP 3-42
- logging firewall events 17-8, 17-21
- notifications of firewall events 17-21
- PBX debugging 14-18
- PPP interfaces 3-42
 - debugging 3-43
- PPP templates
 - debugging 3-43
- tests 9-8

displaying

- activity counters on PPP interfaces 3-17
- aliases 1-58
- calls, PBX 14-26
- clock 1-80
- configuration, PBX 14-25
- configuration, PPP 3-59
- counters
 - ENCO 8-13
- counters, PPP 3-65
- date 1-80
- debugging, PPP 3-75
- directory of FLASH files 1-17
- domain name 4-27, 4-108
- dynamic configuration 1-61
- extensions, PBX 14-27
- files 1-67
- firewall policies 17-4 to 17-5, 17-26
- firewall rules 17-5, 17-26
- firewall status 17-3, 17-25

- FLASH files 1-65, 1-67
- groups, PBX 14-29
- hardware test results 9-10
- install information 1-74
- interfaces table, MIB-II 2-11, 2-30
- interfaces, PPP 3-58
- IP address pools 6-124
- memory contents 1-35
- multilink information, PPP 3-78
- nameservers, PPP 3-80
- PBX calls 14-26
- PBX configuration 14-25
- PBX extensions 14-27
- PBX groups 14-29
- PPP configuration 3-59
- PPP counters 3-65
- PPP debugging 3-75
- PPP interfaces 3-58
- PPP multilink information 3-78
- PPP nameservers 3-80
- PPP templates 3-80
- PPP timers 3-76
- PPP transmission queue 3-84
- route templates 6-135
- semipermanent manager port 1-77
- system clock 1-80
- system date 1-80
- system information 1-80
- system time 1-80
- templates 3-80
- test results 9-10
- time 1-80
- timers, PPP 3-76
- transmission queue, PPP 3-84

DNS relay agent 6-21, 6-73, 6-78

domain name

- adding 4-27, 4-50
- deleting 4-27, 4-54
- displaying 4-27, 4-108
- modifying 4-80
- used with ISDN calls 4-27

dot3 MIB object C-13

Dotted notation 16-4

DOV. See data over voice

downloading

- files 1-16, 1-42, 1-47, 1-51, 1-75, 1-82
- patches 1-22
- software releases 1-22

dump

- memory 1-35

DUMP command 1-35

dynamic configuration

- displaying 1-61
- saving 1-30

Dynamic Host Configuration Protocol (DHCP)

15-1 to 15-26

adding

- policy options 15-4
- ranges 15-9

address allocation 15-2



- automatic 15-2
- dynamic 15-2
- static 15-2
- automatic address allocation 15-2
- BOOTP 15-2 to 15-3
- clients 15-23
- commands
 - ADD DHCP POLICY 15-4
 - ADD DHCP RANGE 15-9
 - CREATE DHCP POLICY 15-9
 - CREATE DHCP RANGE 15-10
 - DELETE DHCP POLICY 15-11
 - DELETE DHCP RANGE 15-15
 - DESTROY DHCP POLICY 15-15
 - DESTROY DHCP RANGE 15-16
 - DISABLE DHCP 15-16
 - ENABLE DHCP 15-17
 - SET DHCP POLICY 15-17
 - SHOW DHCP 15-22
 - SHOW DHCP CLIENT 15-23
 - SHOW DHCP POLICY 15-24
 - SHOW DHCP RANGE 15-25
- configuring 15-3
- creating
 - policies 15-9
 - ranges 15-10
- deleting
 - policy options 15-11
 - ranges 15-15
- destroying
 - policies 15-15
 - ranges 15-16
- disabling server 15-16
- displaying
 - clients 15-23
 - policies 15-24
 - ranges 15-25
 - status 15-22
- dynamic address allocation 15-2
- enabling server 15-17
- example configuration 15-3
- INFINITY lease time 15-2
- lease time for IP addresses 15-2
- modifying policy options 15-17
- policies
 - adding options 15-4
 - creating 15-9
 - deleting options 15-11
 - destroying 15-15
 - displaying 15-24
 - modifying options 15-17
- policy 15-2
- range 15-2
- ranges
 - adding 15-9
 - creating 15-10
 - deleting 15-15
 - destroying 15-16
 - displaying 15-25
- reuse IP addresses 15-2

- server 15-2
- static address allocation 15-2
- status 15-22

E

- E1/T1. See Time Division Multiplexing (TDM)
- EDIT command 1-37, 13-3
- editing
 - files 1-16, 1-37
- editor 1-18, 1-37
- EGP group, MIB-II MIB C-11
- egpNeighEventTrigger* MIB object C-12
- ENABLE BOOTP RELAY command 6-77
- ENABLE BRI CTEST command 4-60
- ENABLE BRI DEBUG command 4-61
- ENABLE BRI TEST command 4-62
- ENABLE DHCP command 15-17
- ENABLE ENCO COMPSTATISTICS command 8-6
- ENABLE ENCO DEBUGGING command 8-7
- ENABLE FIREWALL command 17-22
- ENABLE FIREWALL POLICY command 17-22
- ENABLE HTTP DEBUG command 1-40
- ENABLE HTTP SERVER command 1-40
- ENABLE INTERFACE LINKTRAP command 2-14
- ENABLE IP command 6-77
- ENABLE IP DEBUG command 6-78
- ENABLE IP DNSRELAY command 6-78
- ENABLE IP ECHOREPLY command 6-78
- ENABLE IP FOFILTER command 6-78
- ENABLE IP FORWARDING command 6-79
- ENABLE IP HELPER command 6-80
- ENABLE IP INTERFACE command 6-80
- ENABLE IP REMOTEASSIGN command 6-81
- ENABLE IP ROUTE command 6-81
- ENABLE IP SRCROUTE command 6-82
- ENABLE ISDN CALL command 4-65
- ENABLE ISDN LOG command 4-65
- ENABLE LOG command 12-22
- ENABLE LOG GENERATION command 12-23
- ENABLE LOG OUTPUT command 12-23
- ENABLE LOG RECEPTION command 12-23
- ENABLE MIOX CIRCUIT command 5-17
- ENABLE PBX DEBUG command 14-19
- ENABLE PORT command 2-15
- ENABLE PPP command 3-44
- ENABLE PPP DEBUG command 3-45
- ENABLE PPP TEMPLATE DEBUG command 3-46
- ENABLE Q931 ASPID command 4-66
- ENABLE Q931 DEBUG command 4-66
- ENABLE RAPI command 4-71
- ENABLE SNMP AUTHENTICATE_TRAP command 16-18
- ENABLE SNMP command 16-17
- ENABLE SNMP COMMUNITY command 16-18
- ENABLE TEST INTERFACE command 9-9
- ENABLE TRIGGER command 10-12
- ENABLE USER command 1-41
- enabling
 - debugging
 - ENCO 8-7
 - PBX 14-19



- debugging, PPP interfaces 3-45
- debugging, PPP templates 3-46
- ENCO debugging 8-7
- firewall 17-3, 17-22
- firewall accounting 17-22
- firewall debugging 17-6, 17-22
- hardware tests 9-9
- interfaces, PPP 3-44
- logging firewall events 17-8, 17-22
- PBX debugging 14-19
- PPP interfaces 3-44
 - debugging 3-45
- PPP templates
 - debugging 3-46
- tests 9-9
- encapsulation 2-2
 - 802.2 2-4
 - Ethernet 2-4
 - PPP 3-3
 - SNAP 2-4
- encapsulations
 - X.25 5-4
- ENCO. See Compression and Encryption Services
- encryption
 - Point-to-Point Protocol (PPP) 3-25
- encryption. See Compression and Encryption Services; Link Compression and Encryption; IP Security (IPsec)
- Endpoint Discriminator Option, LCP 3-5
- entering commands 1-4
- enterprise MIB. See Management Information Base C-3
- Ethernet 2-3
 - commands
 - RESET ETH 2-16
 - RESET ETH COUNTERS 2-16
 - SHOW ETH CONFIGURATION 2-22
 - SHOW ETH COUNTERS 2-23
 - SHOW ETH MACADDRESS 2-29
 - SHOW ETH RECEIVE 2-29
 - configuration 2-16, 2-22
 - counters 2-16, 2-23
 - encapsulation 2-4
 - interface on router 2-3
 - MIB 2-23
 - multicast addresses 2-6, 2-29
 - reset 2-16
- Ethernet Group, Allied Telesyn Enterprise MIB C-7
- ethernet* MIB object C-7
- Ethernet-like Collision Statistics group, Ethernet-like Interface Types MIB C-13
- Ethernet-like Statistics group, Ethernet-like Interface Types MIB C-13
- ethIntTable* MIB object C-7
- event logging 1-13
 - See also Logging Facility
- events
 - firewall 17-21
- events, Point-to-Point Protocol (PPP) 3-4
- example configurations
 - AODI 4-29
 - DHCP 15-3

- firewall 17-8 to 17-11
- Logging Facility 12-9
- Point-to-Point Protocol (PPP) 3-20
- SNMP 16-11
- Time Division Multiplexing (TDM) 11-2
- examples
 - installing a compressed release 1-25
 - installing a standard release 1-24
 - Internet Protocol (IP) configuration 6-33
 - patches 1-24
 - software releases 1-24
- exceptions 1-13, 1-64
- exclusion filters
 - Internet Protocol (IP) 6-45, 6-64, 6-88, 6-117
- expansion options
 - testing 9-1 to 9-12
- extensions
 - creating PBX 14-13
 - destroying PBX 14-17
 - modifying PBX 14-21

F

- fatal errors 1-13, 1-64
- fault diagnosis 1-13
- File Group, Allied Telesyn Enterprise MIB C-10
- file* MIB object C-10
- file naming conventions 1-15
 - using wildcards 1-16
- file subsystem 1-15
 - creating files 1-16
 - deleting files 1-16, 1-32
 - displaying directory 1-16
 - displaying files 1-67
 - downloading files 1-16, 1-42, 1-47, 1-51, 1-75, 1-82
 - editing files 1-16, 1-37
 - file types 1-15
 - naming conventions 1-15
 - renaming files 1-46
 - wildcards 1-16
- file types 1-15
- files
 - creating 1-16
 - deleting 1-16, 1-32
 - displaying 1-67
 - downloading 1-16, 1-42, 1-47, 1-51, 1-75, 1-82
 - editing 1-16, 1-37
 - renaming 1-46
- fileTable* MIB object C-10
- filters
 - exclusion
 - Internet Protocol (IP) 6-45, 6-64, 6-88, 6-117
 - fragment offset
 - Internet Protocol (IP) 6-73, 6-78
 - inclusion
 - Internet Protocol (IP) 6-45, 6-64, 6-88, 6-117
- Internet Protocol (IP) 6-73, 6-78
 - exclusion 6-45, 6-64, 6-88, 6-117
 - fragment offset 6-73, 6-78
 - inclusion 6-45, 6-64, 6-88, 6-117
 - routing information 6-59, 6-69, 6-100, 6-134



- routing information
 - Internet Protocol (IP) 6-13, 6-59, 6-69, 6-100, 6-134
- traffic
 - Internet Protocol (IP) 6-21
- Firewall 17-1 to 17-33
 - access lists
 - adding 17-14
 - deleting 17-19
 - accounting
 - disabling 17-21
 - enabling 17-22
 - adding
 - access lists 17-14
 - interfaces 17-4, 17-11
 - NAT translations 17-6, 17-12
 - RADIUS servers 17-14
 - rules 17-4, 17-14
 - application gateway 17-2
 - commands 17-11 to 17-33
 - ADD FIREWALL POLICY INTERFACE 17-11
 - ADD FIREWALL POLICY NAT 17-12
 - ADD FIREWALL POLICY RULE 17-14
 - CREATE FIREWALL POLICY 17-16
 - DELETE FIREWALL POLICY INTERFACE 17-17
 - DELETE FIREWALL POLICY NAT 17-18
 - DELETE FIREWALL POLICY RULE 17-19
 - DELETE FIREWALL SESSION 17-19
 - DESTROY FIREWALL POLICY 17-20
 - DISABLE FIREWALL 17-20
 - DISABLE FIREWALL NOTIFY 17-21
 - DISABLE FIREWALL POLICY 17-21
 - ENABLE FIREWALL 17-22
 - ENABLE FIREWALL POLICY 17-22
 - SET FIREWALL POLICY RULE 17-24
 - SHOW FIREWALL 17-25
 - SHOW FIREWALL POLICY 17-26
 - SHOW FIREWALL POLICY SESSION 17-32
 - configuration examples 17-8 to 17-11
 - creating policies 17-3, 17-16
 - debugging 17-6
 - disabling 17-6, 17-21
 - enabling 17-6, 17-22
 - deleting
 - access lists 17-19
 - interfaces 17-4, 17-17
 - NAT translations 17-6, 17-18
 - rules 17-4, 17-19
 - destroying policies 17-3, 17-20
 - disabling 17-3, 17-20
 - accounting 17-21
 - debugging 17-6, 17-21
 - event logging 17-8, 17-21
 - ICMP forwarding 17-5, 17-21
 - IP options 17-5, 17-21
 - notifications of events 17-21
 - ping 17-5, 17-21
 - displaying
 - policies 17-4 to 17-5, 17-26
 - rules 17-5, 17-26
 - status 17-3, 17-25
 - dynamic packet filtering 17-2
 - enabling 17-3, 17-22
 - accounting 17-22
 - debugging 17-6, 17-22
 - event logging 17-8, 17-22
 - ICMP forwarding 17-5, 17-22
 - IP options 17-5, 17-22
 - ping 17-5, 17-22
 - events
 - logging 17-6
 - notifying 17-21
 - example configurations 17-8 to 17-11
 - feature summary 17-2
 - ICMP forwarding
 - disabling 17-5, 17-21
 - enabling 17-5, 17-22
 - IP options
 - disabling 17-5, 17-21
 - enabling 17-5, 17-22
 - logging events 17-6
 - disabling 17-8, 17-21
 - enabling 17-8, 17-22
 - modifying
 - rules 17-4, 17-24
 - monitoring 17-6 to 17-8
 - NAT 17-6
 - adding translations 17-6, 17-12
 - deleting translations 17-6, 17-18
 - notifications
 - disabling 17-21
 - ping
 - disabling 17-5, 17-21
 - enabling 17-5, 17-22
 - policies 17-3 to 17-4
 - adding interfaces 17-4, 17-11
 - creating 17-3, 17-16
 - deleting interfaces 17-4, 17-17
 - destroying 17-3, 17-20
 - displaying 17-4 to 17-5, 17-26
 - RADIUS servers
 - adding 17-14
 - rules 17-4 to 17-5
 - adding 17-4, 17-14
 - deleting 17-4, 17-19
 - displaying 17-5, 17-26
 - modifying 17-4, 17-24
 - security policies 17-3 to 17-4
 - stateful inspection 17-2
- Firewall Group, Allied Telesyn Enterprise MIB C-10
- firewall* MIB object C-10
- firewallTrap* trap message C-10
- firewallTrapMessage* MIB object C-10
- FLASH File System (FFS)
 - creating files 1-31
 - deleting files 1-32
 - displaying files 1-65
- FLASH files 1-16
 - creating 1-31
 - deleting 1-32
 - displaying 1-65, 1-67



- displaying directory 1-17
- downloading 1-42, 1-47, 1-51, 1-75, 1-82
- editing 1-37
- renaming 1-46
- FLASH memory 1-14
 - clearing 1-29
 - compacting 1-17, 1-26
 - displaying 1-15, 1-67, 1-69
- FLUSH LOG OUTPUT command 12-24
- format of file names 1-15 to 1-16
- format, SNMP messages 16-6
- fragment offset filter
 - Internet Protocol (IP) 6-73, 6-78
- framework, for network management 16-2

G

- get-next PDU 16-6
- get-request PDU 16-6
- get-response PDU 16-6
- groups
 - creating PBX 14-16
 - destroying PBX 14-18
 - modifying PBX 14-24
- Groups, MIB
 - address translation C-11
 - arInterfaces C-6
 - BRI C-8
 - CMOT C-11
 - EGP C-11
 - Ethernet C-7
 - Ethernet-like Collision Statistics C-13
 - Ethernet-like Statistics C-13
 - File C-10
 - File Group C-10
 - Host Resources Device C-14 to C-15
 - Host Resources Installed Software C-14 to C-16
 - Host Resources Running Software C-14 to C-15
 - Host Resources Running Software Performance C-14 to C-16
 - Host Resources Storage C-14 to C-15
 - Host Resources System C-14 to C-15
 - ICMP C-11
 - Install C-9
 - interfaces C-11
 - IP C-11
 - ISDN Call Control C-8
 - Loader C-9
 - Modules C-7
 - Objects C-4
 - PRI C-9
 - SNMP C-12
 - system C-11
 - TCP C-11
 - Transmission C-12 to C-13
 - UDP C-11

H

- hardware
 - testing 9-1 to 9-12
- header compression 8-3

- header, Internet Protocol (IP) 6-4
- help
 - displaying 1-5, 1-41
 - setting system help file 1-50
- HELP command 1-41
- history, command recall 1-4
- history, compression 8-2
- host
 - adding Internet Protocol (IP) 6-52
 - changing Internet Protocol (IP) 6-91
 - deleting Internet Protocol (IP) 6-65
 - displaying Internet Protocol (IP) 6-120
 - modifying Internet Protocol (IP) 6-91
 - setting Internet Protocol (IP) 6-91
- host MIB object C-14
- Host Resources Device group, Host Resources MIB C-14 to C-15
- Host Resources Installed Software group, Host Resources MIB C-14 to C-15
- Host Resources Running Software group, Host Resources MIB C-14 to C-15
- Host Resources Running Software Performance group, Host Resources MIB C-14 to C-15
- Host Resources Storage group, Host Resources MIB C-14 to C-15
- Host Resources System group, Host Resources MIB C-14 to C-15
- hrDeviceDiskStorage MIB object C-15
- hrDeviceEntry MIB object C-15
- hrDeviceID MIB object C-4, C-15
- hrDeviceType MIB object C-15
- hrDiskStorageTable MIB object C-15
- hrFSTable MIB object C-15
- hrNetworkTable MIB object C-15
- hrPrinterTable MIB object C-15
- hrStorageAllocationUnits MIB object C-15
- hrSWRunTable MIB object C-14
- hrSystemDate MIB object C-15
- hrSystemInitialLoadDevice MIB object C-15
- hrSystemInitialLoadParameters MIB object C-15
- hrSystemMaxProcesses MIB object C-15
- hrSystemNumUsers MIB object C-15
- hrSystemProcesses MIB object C-15
- HTTP Server ?? to 1-20
 - resolving URLs 1-20
- HTTP server 1-19
 - disabling 1-34
 - displaying client status 1-69
 - displaying debugging status 1-70
 - displaying session information 1-71
 - displaying status 1-72
 - enabling 1-40
 - enabling debugging 1-40
 - resetting 1-47
- hunting, PBX 14-16, 14-18, 14-24

I

- ICMP group, MIB-II MIB C-11
- identifier, instance 16-4
- identifier, MIB object 16-4

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

- combined notation 16-4
- dotted notation 16-4
- textual notation 16-4
- IEEE 802.2, Ethernet encapsulation 2-4
- if..then..else..endif, in scripts 13-4, 13-8
- iface_asyn* MIB object C-5
- iface_bri* MIB object C-5
- iface_eth* MIB object C-5
- iface_pots* MIB object C-5
- iface_pri* MIB object C-5
- iface_syn* MIB object C-5
- ifAdminStatus* MIB object C-12
- ifIndex* MIB object C-7 to C-9
- ifSpecific* MIB object C-13
- ifType* MIB object C-13
- ifTypes* MIB object C-5
- inclusion filters
 - Internet Protocol (IP) 6-45, 6-64, 6-88, 6-117
- Install Group, Allied Telesyn Enterprise MIB C-9
- install information 1-23
 - default install 1-23
 - deleting 1-33
 - displaying 1-74
 - preferred install 1-23
 - setting 1-50
 - temporary install 1-23
- install* MIB object C-9
- Installed Software Group, Host Resources MIB C-16
- installHistoryTable* MIB object C-9
- installTable* MIB object C-9
- instance 2-2
- instance identifier, MIB object 16-4
- instance, MIB object 16-4
- Integrated Services Digital Network (ISDN) 4-1
 - adding domain name 4-27, 4-50
 - addressing, LAPD 4-13
 - AODI 4-28 to 4-31
 - components 4-28
 - configuration example 4-29
 - automatic switch detection 4-21
 - auto-SPIDs 4-18, 4-43, 4-66
 - Basic Rate Access 4-3 to 4-6
 - configuring 4-7
 - BRI physical layer 4-7, 4-56 to 4-57, 4-60 to 4-62, 4-72 to 4-73, 4-86, 4-88, 4-94 to 4-96, 4-100
 - call control 4-22, 4-42, 4-44, 4-50, 4-52 to 4-53, 4-58, 4-65, 4-75, 4-103, 4-107, 4-112
 - call logging 4-27, 4-58, 4-65, 4-81, 4-109
 - CAPI 4-26, 4-60, 4-71
 - commands
 - ACTIVATE ISDN CALL 4-42, 4-112
 - ACTIVATE Q931 ASPID 4-43
 - ADD ISDN CALL 4-44
 - ADD ISDN CLILIST 4-50
 - ADD ISDN DOMAINNAME 4-50
 - ADD LAPD 4-82
 - ADD LAPD TEI 4-51
 - ADD LAPD XSPID 4-51
 - ADD LAPD XTEI 4-52
 - DEACTIVATE ISDN CALL 4-52
 - DELETE ISDN CALL 4-53
 - DELETE ISDN CLILIST 4-53
 - DELETE ISDN DOMAINNAME 4-54
 - DELETE LAPD TEI 4-54
 - DELETE LAPD XSPID 4-55
 - DELETE LAPD XTEI 4-55
 - DISABLE BRI CTEST 4-56
 - DISABLE BRI DEBUG 4-56
 - DISABLE BRI TEST 4-57
 - DISABLE ISDN CALL 4-58
 - DISABLE ISDN LOG 4-58
 - DISABLE Q931 DEBUG 4-59
 - DISABLE RCAP 4-60
 - ENABLE BRI CTEST 4-60
 - ENABLE BRI DEBUG 4-61
 - ENABLE BRI TEST 4-62
 - ENABLE ISDN CALL 4-65
 - ENABLE ISDN LOG 4-65
 - ENABLE Q931 ASPID 4-66
 - ENABLE Q931 DEBUG 4-66
 - ENABLE RCAP 4-71
 - RESET BRI 4-72
 - RESET BRI COUNTERS 4-72
 - RESET Q931 4-73
 - SET BRI 4-73
 - SET ISDN CALL 4-75
 - SET ISDN DOMAINNAME 4-80
 - SET ISDN LOG 4-81
 - SET Q931 4-84
 - SHOW BRI CONFIGURATION 4-86
 - SHOW BRI COUNTERS 4-88
 - SHOW BRI CTEST 4-94
 - SHOW BRI DEBUG 4-95
 - SHOW BRI STATE 4-96
 - SHOW BRI TEST 4-100
 - SHOW ISDN CALL 4-103
 - SHOW ISDN CLILIST 4-107
 - SHOW ISDN DOMAINNAME 4-108
 - SHOW ISDN LOG 4-109
 - SHOW LAPD 4-110
 - SHOW LAPD COUNT 4-112
 - SHOW LAPD STATE 4-114
 - SHOW Q931 4-114
 - SHOW Q931 SPID 4-117
 - common D channel signalling 4-15
 - configuring 4-33
 - Basic Rate Access 4-7
 - counters 4-72, 4-88, 4-112
 - data over voice 4-32
 - debugging
 - SPIDs 4-19
 - deleting domain name 4-27, 4-54
 - Device Control Protocol (DCP) 4-26, 4-60, 4-71
 - displaying domain name 4-27, 4-108
 - domain name 4-27
 - adding 4-27, 4-50
 - deleting 4-27, 4-54
 - displaying 4-27, 4-108
 - modifying 4-80
 - interfaces, slotted 4-28



- LAPD 4-11, 4-51, 4-54, 4-82, 4-110, 4-114
- LAPD addressing 4-13
- modifying domain name 4-80
- non-associated signalling 4-15
- packet mode 4-12, 4-51 to 4-52, 4-55
- profile 4-16
- profiles 4-17 to 4-22, 4-117
- Q.931 4-15, 4-59, 4-66, 4-73, 4-84, 4-114
- Remote CAPI 4-26, 4-60, 4-71
- S/T interfaces 4-4
- service provider profiles 4-17 to 4-22, 4-117
- slotted interfaces 4-28
- SPIDs 4-17 to 4-22, 4-117
 - automatic selection 4-43, 4-66
 - debugging 4-19
- support 4-6
- switch detection 4-21
- telephony services 14-2
- territory 4-16, 4-33, 4-35
- X.25 over LAPD 4-12, 4-51 to 4-52, 4-55
- Integrated Services Digital Network (ISDN)U interfaces 4-6
- Integrated Services Digital Network ISDN)
 - SPIDs
 - automatic selection 4-18
- Interfaces 2-1 to 2-39
 - link up/down traps 2-12 to 2-14, 2-18
 - managing with SNMP 2-12
- interfaces
 - adding
 - Internet Protocol (IP) 6-53
 - PPP physical links 3-28
 - asynchronous ports 2-7
 - changing
 - Internet Protocol (IP) 6-94
 - commands
 - DISABLE INTERFACE LINKTRAP 2-13
 - ENABLE INTERFACE LINKTRAP 2-14
 - SET INTERFACE TRAPLIMIT 2-18
 - SHOW INTERFACE 2-30
 - configuration
 - asynchronous ports 2-7
 - Ethernet 2-16
 - creating
 - PPP 3-31
 - defined 2-2
 - deleting
 - Internet Protocol (IP) 6-66
 - PPP physical links 3-41
 - destroying
 - PPP 3-41
 - disabling
 - Internet Protocol (IP) 6-75
 - PPP 3-42
 - displaying
 - Internet Protocol (IP) 6-121
 - PPP 3-58
 - displaying MIB 2-11, 2-30
 - enabling
 - Internet Protocol (IP) 6-80
 - PPP 3-44
- Ethernet 2-16
- instance 2-2
- Internet Protocol (IP)
 - adding 6-53
 - changing 6-94
 - deleting 6-66
 - disabling 6-75
 - displaying 6-121
 - enabling 6-80
 - modifying 6-92, 6-94
 - resetting 6-85
 - setting 6-94
- modifying
 - Internet Protocol (IP) 6-92, 6-94
 - PPP 3-49
- naming 2-2
- PPP
 - adding physical links 3-28
 - creating 3-31
 - deleting physical links 3-41
 - destroying 3-41
 - disabling 3-42
 - displaying 3-58
 - enabling 3-44
 - modifying 3-49
 - purging 3-47
 - resetting 3-48
- purging
 - PPP 3-47
- resetting
 - Internet Protocol (IP) 6-85
 - PPP 3-48
- setting
 - Internet Protocol (IP) 6-94
- testing 9-1 to 9-12
- types 2-2
- interfaces group, MIB-II C-11
- Internet 6-4
- Internet Control Message Protocol (ICMP) 6-11
- Internet Protocol (IP)
 - adding
 - ARP entries 6-10, 6-44
 - BOOTP relay destination 6-43
 - exclusion filters 6-45
 - filters 6-45
 - hosts 6-52
 - inclusion filters 6-45
 - interfaces 6-53
 - metric boosts 6-16
 - proxy ARPs 6-10
 - RIP addresses 6-15, 6-56
 - route templates 6-60
 - routes 6-13, 6-56 to 6-57
 - routing filters 6-59
 - trusted routers 6-61
 - address assignment
 - remote 6-32, 6-75, 6-81
 - using DHCP 6-11, 6-73, 6-78
 - address classes 6-6
 - address format 6-6



- address pools 6-32
 - creating 6-62
 - destroying 6-71
 - displaying 6-124
- Address Resolution Protocol (ARP)
 - adding entries 6-10, 6-44
 - changing entries 6-86
 - deleting entries 6-63
 - displaying entries 6-10, 6-108
 - modifying entries 6-86
 - proxy 6-10
 - setting entries 6-86
- addresses 6-4, 6-6, 6-32, 6-75, 6-81
- agent, SNMP
 - disabling 16-16
 - displaying 16-20
 - enabling 16-17
- authentication trap, SNMP
 - disabling 16-16
 - enabling 16-18
- authentication, SNMP 16-10
- autonomous system number 6-87
- boosts 6-16
- BOOTP relay agent 6-29, 6-43, 6-63, 6-71, 6-77, 6-105
- broadcast forwarding 6-26 to 6-29, 6-51, 6-64, 6-74, 6-80, 6-119
 - to a broadcast address 6-28
 - to a unicast address 6-27
- changing
 - ARP entries 6-86
 - hosts 6-91
 - interfaces 6-94
 - RIP addresses 6-96
 - RIP timers 6-97
 - routes 6-96, 6-98
- commands 6-43, 16-12
 - ADD BOOTP RELAY 6-43
 - ADD IP ARP 6-44
 - ADD IP FILTER 6-45
 - ADD IP HELPER 6-51
 - ADD IP HOST 6-52
 - ADD IP INTERFACE 6-53
 - ADD IP RIP 6-56
 - ADD IP ROUTE 6-57
 - ADD IP ROUTE FILTER 6-59
 - ADD IP ROUTE TEMPLATE 6-60
 - ADD IP TRUSTED 6-61
 - ADD SNMP COMMUNITY 16-13
 - CREATE IP POOL 6-62
 - CREATE SNMP COMMUNITY 16-14
 - DELETE BOOTP RELAY 6-63
 - DELETE IP ARP 6-63
 - DELETE IP FILTER 6-64
 - DELETE IP HELPER 6-64
 - DELETE IP HOST 6-65
 - DELETE IP INTERFACE 6-66
 - DELETE IP RIP 6-67
 - DELETE IP ROUTE 6-68
 - DELETE IP ROUTE FILTER 6-69
 - DELETE IP ROUTE TEMPLATE 6-69
 - DELETE IP TRUSTED 6-69
 - DELETE SNMP COMMUNITY 16-15
 - DELETE TCP 6-70
 - DESTROY IP POOL 6-71
 - DESTROY SNMP COMMUNITY 16-16
 - DISABLE BOOTP RELAY 6-71
 - DISABLE IP 6-72
 - DISABLE IP DEBUG 6-72
 - DISABLE IP DNSRELAY 6-73
 - DISABLE IP ECHOREPLY 6-73
 - DISABLE IP FOFILTER 6-73
 - DISABLE IP FORWARDING 6-74
 - DISABLE IP HELPER 6-74
 - DISABLE IP INTERFACE 6-75
 - DISABLE IP REMOTEASSIGN 6-75
 - DISABLE IP ROUTE 6-76
 - DISABLE IP SRCROUTE 6-76
 - DISABLE SNMP 16-16
 - DISABLE SNMP AUTHENTICATE_TRAP 16-16
 - DISABLE SNMP COMMUNITY 16-17
 - ENABLE BOOTP RELAY 6-77
 - ENABLE IP 6-77
 - ENABLE IP DEBUG 6-78
 - ENABLE IP DNSRELAY 6-78
 - ENABLE IP ECHOREPLY 6-78
 - ENABLE IP FOFILTER 6-78
 - ENABLE IP FORWARDING 6-79
 - ENABLE IP HELPER 6-80
 - ENABLE IP INTERFACE 6-80
 - ENABLE IP REMOTEASSIGN 6-81
 - ENABLE IP ROUTE 6-81
 - ENABLE IP SRCROUTE 6-82
 - ENABLE SNMP 16-17
 - ENABLE SNMP AUTHENTICATE_TRAP 16-18
 - ENABLE SNMP COMMUNITY 16-18
 - PING 6-82
 - PURGE BOOTP RELAY 6-84
 - PURGE IP 6-84
 - RESET IP 6-84
 - RESET IP COUNTER 6-85
 - RESET IP INTERFACE 6-85
 - SET BOOTP MAXHOPS 6-86
 - SET IP ARP 6-86
 - SET IP AUTONOMOUS 6-87
 - SET IP FILTER 6-88
 - SET IP HOST 6-91
 - SET IP INTERFACE 6-92
 - SET IP LOCAL 6-94
 - SET IP NAMESERVER 6-95
 - SET IP RIP 6-96
 - SET IP RIPTIMER 6-97
 - SET IP ROUTE 6-98
 - SET IP ROUTE FILTER 6-100
 - SET IP ROUTE TEMPLATE 6-101
 - SET IP SECONDARYNAMESERVER 6-102
 - SET PING 6-103
 - SET SNMP COMMUNITY 16-19
 - SET TRACE 6-104
 - SHOW BOOTP RELAY 6-105
 - SHOW IP 6-106



- SHOW IP ARP 6-108
- SHOW IP COUNTER 6-109
- SHOW IP DEBUG 6-116
- SHOW IP FILTER 6-117
- SHOW IP HELPER 6-119
- SHOW IP HOST 6-120
- SHOW IP INTERFACE 6-121
- SHOW IP POOL 6-124
- SHOW IP RIP 6-126
- SHOW IP RIP COUNTER 6-128
- SHOW IP RIPTIMER 6-127
- SHOW IP ROUTE 6-130
- SHOW IP ROUTE FILTER 6-134
- SHOW IP ROUTE TEMPLATE 6-135
- SHOW IP TRUSTED 6-136
- SHOW IP UDP 6-136
- SHOW PING 6-137
- SHOW SNMP 16-20
- SHOW SNMP COMMUNITY 16-22
- SHOW TCP 6-139
- SHOW TRACE 6-143
- STOP PING 6-145
- STOP TRACE 6-145
- TRACE 6-146
- communities, SNMP 16-9
 - adding 16-13
 - creating 16-14
 - deleting 16-15
 - destroying 16-16
 - disabling 16-17
 - displaying 16-22
 - enabling 16-18
 - modifying 16-19
- configuration 6-33, 6-106
 - examples 6-33
- connection vs. connectionless mode 6-4
- control functions 6-24
- counters 6-109, 6-128
 - resetting 6-85
- creating
 - address pools 6-62
- debugging functions 6-24 to 6-25, 6-72, 6-78, 6-116
- definition 6-4
- deleting
 - ARP entries 6-63
 - BOOTP relay destination 6-63
 - exclusion filters 6-64
 - filters 6-64
 - hosts 6-65
 - inclusion filters 6-64, 6-88
 - interfaces 6-66
 - metric boosts 6-16
 - RIP addresses 6-67
 - route templates 6-69
 - routes 6-13, 6-67 to 6-68
 - routing filters 6-69
 - TCP sessions 6-70
 - trusted routers 6-69
- destination address 6-4
- destroying
 - address pools 6-71
- DHCP client 6-11, 6-54, 6-73, 6-78, 6-93
- disabling
 - BOOTP relay agent 6-71
 - fragment offset filter 6-73
 - interfaces 6-75
 - routes 6-76
- displaying
 - address pools 6-124
 - ARP entries 6-10, 6-35, 6-108
 - BOOTP relay destination 6-105
 - configuration 6-106
 - counters 6-109, 6-128
 - exclusion filters 6-117
 - hosts 6-120
 - inclusion filters 6-117
 - interfaces 6-121
 - metric boosts 6-16
 - proxy ARPs 6-10
 - RIP addresses 6-15, 6-126
 - RIP counters 6-128
 - RIP timers 6-127
 - route templates 6-135
 - routes 6-13, 6-35, 6-130
 - routing filters 6-134
 - TCP sessions 6-139
 - trusted routers 6-136
 - UDP sessions 6-136
- DNS relay agent 6-21, 6-73, 6-78
- echo reply
 - disabling 6-73
 - enabling 6-78
- enabling
 - BOOTP relay agent 6-77
 - fragment offset filter 6-78
 - interfaces 6-80
 - routes 6-81
- examples 6-33
- exclusion filters 6-45, 6-64, 6-88, 6-117
- filters 6-13, 6-21, 6-45, 6-64, 6-73, 6-78, 6-88, 6-117
- fragment offset filter 6-73, 6-78
- gateway 6-35, 6-72, 6-74, 6-77, 6-79, 6-106
- header 6-4
- hosts 6-52, 6-65, 6-91, 6-120
- inclusion filters 6-45, 6-64, 6-88, 6-117
- interfaces
 - adding 6-53
 - changing 6-94
 - deleting 6-66
 - disabling 6-75
 - displaying 6-121
 - enabling 6-80
 - modifying 6-92, 6-94
 - resetting 6-85
 - setting 6-94
- Internet Control Message Protocol (ICMP) 6-11
- local interface 6-94
- metrics 6-16
- MIB 6-23, 6-109, 16-1
- modifying



- ARP entries 6-86
- hosts 6-91
- interfaces 6-92, 6-94
- route templates 6-101
- routes 6-98
- routing filters 6-100
- multicasting 6-31
- multihoming 6-9
- name server 6-95, 6-102
- ping function 6-25, 6-82, 6-103, 6-137, 6-145
- PING utility 6-36
- primary name server 6-95
- proxy ARPs 6-10
- purge configuration database 6-84
- reinitialise module 6-84
- remote address assignment 6-32, 6-75, 6-81
- resetting
 - counters 6-85
 - interfaces 6-85
- restart module 6-84
- RIP timers
 - displaying 6-127
 - setting 6-97
- route templates 6-19
 - adding 6-60
 - deleting 6-69
 - displaying 6-135
 - modifying 6-101
- router type 6-35, 6-72, 6-74, 6-77, 6-79
- routes 6-12
 - adding 6-13, 6-57
 - changing 6-98
 - deleting 6-68
 - disabling 6-76
 - displaying 6-13, 6-130
 - enabling 6-81
 - modifying 6-98
 - setting 6-98
 - tracing 6-25, 6-104, 6-143, 6-145 to 6-146
- routing 6-12
- routing filters 6-59
 - deleting 6-69
 - displaying 6-134
 - modifying 6-100
- routing information filters 6-13
- Routing Information Protocol (RIP) 6-14, 6-56, 6-67, 6-96 to 6-97, 6-126 to 6-128
- routing protocols
 - Routing Information Protocol (RIP) 6-14, 6-56, 6-67, 6-96 to 6-97, 6-127
- routing table 6-12, 6-130
- secondary name server 6-102
- security options 6-26, 6-73, 6-78
- server 6-35, 6-72, 6-74, 6-77, 6-79, 6-106
- setting
 - ARP entries 6-86
 - displaying 6-117
 - exclusion filters 6-88
 - filters 6-88
 - hosts 6-91

- interfaces 6-94
 - routes 6-98
- SNMP 6-23, 16-1
- SNMP agent
 - disabling 16-16
 - displaying 16-20
 - enabling 16-17
- SNMP authentication 16-10
- SNMP authentication trap
 - disabling 16-16
 - enabling 16-18
- SNMP communities 16-9
 - adding 16-13
 - creating 16-14
 - deleting 16-15
 - destroying 16-16
 - disabling 16-17
 - displaying 16-22
 - enabling 16-18
 - modifying 16-19
- source address 6-4
- subnet 6-8
- subnet mask 6-8
- TCP sessions 6-70, 6-139
- Telnet server 6-35, 6-72, 6-74, 6-77, 6-79, 6-106
- timers
 - RIP 6-97, 6-127
- trace route 6-25, 6-104, 6-143, 6-145 to 6-146
- traffic filters 6-21
- trusted routers 6-61, 6-69, 6-136
- UDP sessions 6-136
- Internet-standard MIB 16-3
- Internet-standard Network Management Framework 16-2
- IP address 6-54, 6-93
 - from DHCP 6-11, 6-73, 6-78
- IP group, MIB-II MIB C-11
- ipNetToMediaIfIndex* MIB object C-12
- ipNetToMediaNetAddress* MIB object C-12
- ipNetToMediaPhysAddress* MIB object C-12
- ipNetToMediaType* MIB object C-12
- ipRouteAge* MIB object C-12
- ipRouteIfIndex* MIB object C-12
- ipRouteMask* MIB object C-12
- ipRouteMetric1* MIB object C-12
- ipRouteMetric2* MIB object C-12
- ipRouteMetric3* MIB object C-12
- ipRouteMetric4* MIB object C-12
- ipRouteMetric5* MIB object C-12
- ipRouteNextHop* MIB object C-12
- ipRouteType* MIB object C-12
- ISDN Call Control Group, Allied Telesyn Enterprise MIB C-8
- ISDN. See Integrated Services Digital Network (ISDN)

L

- LAPD, ISDN 4-11, 4-51, 4-54, 4-82, 4-110, 4-112, 4-114
 - addressing 4-13
- Least Cost Routing (LCR)
 - tieline dialling 14-20
- Lempel-Ziv compression algorithm 8-2
- licenceTable* MIB object C-10



limits, on PPP interfaces 3-16, 3-31, 3-36, 3-48 to 3-49, 3-54

link compression 8-2

Link Compression and Encryption 18-1 to 18-4
See also Compression and Encryption Services; IP Security (IPsec)

compression, link 18-2 to 18-4

link compression 18-2 to 18-4

overview 18-2

PPP 18-3

user modules

- PPP 18-3
- X.25 18-4
- X.25 18-4

link compression. See Link Compression and Encryption

Link Control Protocol (LCP) 3-3 to 3-4, 3-19

options 3-5

- Authentication Protocol 3-5
- Endpoint Discriminator 3-5
- Link Discriminator 3-5 to 3-6
- Link Quality Reporting (LQR) 3-5 to 3-6
- Magic Number 3-5
- Maximum Receive Unit 3-5
- Maximum Received Reconstructed Unit 3-5

Link Discriminator Option, LCP 3-5 to 3-6

link management, PPP 3-16, 3-31, 3-36, 3-48 to 3-49, 3-54

link quality management (LQM) 3-19, 3-24

Link Quality Reporting (LQR) Option, LCP 3-5 to 3-6

link up/down traps 2-12 to 2-14, 2-18

LOAD command 1-42, 13-3

Loader Group, Allied Telesyn Enterprise MIB C-9

loader MIB object C-9

loadStatus MIB object C-9

loadTable MIB object C-9

location

- displaying 1-78
- setting 1-55

logging

- firewall events 17-6

Logging Facility 12-1

actions 12-5

- adding log filters 12-8, 12-12
- changing log filters 12-9, 12-25
- changing output definitions 12-8, 12-25
- changing reception of log messages 12-29

commands

- ADD LOG OUTPUT 12-12
- ADD LOG RECEIVE 12-15
- CREATE LOG OUTPUT 12-16
- DELETE LOG OUTPUT 12-19
- DELETE LOG RECEIVE 12-20
- DESTROY LOG OUTPUT 12-20
- DISABLE LOG 12-21
- DISABLE LOG GENERATION 12-21
- DISABLE LOG OUTPUT 12-21
- DISABLE LOG RECEPTION 12-22
- ENABLE LOG 12-22
- ENABLE LOG GENERATION 12-23
- ENABLE LOG OUTPUT 12-23

- ENABLE LOG RECEPTION 12-23
- FLUSH LOG OUTPUT 12-24
- PURGE LOG 12-24
- SET LOG OUTPUT 12-25
- SET LOG RECEIVE 12-29
- SET LOG UTCOFFSET 12-30
- SHOW LOG 12-31 to 12-32
- SHOW LOG COUNTERS 12-38
- SHOW LOG OUTPUT 12-40, 12-42
- SHOW LOG QUEUE 12-44, 12-47
- SHOW LOG RECEIVE 12-45
- SHOW LOG STATUS 12-46

comparison operators 12-9, 12-25

configuration example 12-9

configuring 12-46

counters 12-38

creating output definitions 12-8, 12-16

deleting log filters 12-9

deleting output definitions 12-8, 12-19 to 12-20

destinations 12-5 to 12-6, 12-16, 12-24 to 12-25, 12-42

disabling 12-21, 12-46

disabling generation of log messages 12-21

disabling output definitions 12-8, 12-21, 12-40

disabling reception of log messages 12-22

displaying counters 12-38

displaying log filters 12-9, 12-40

displaying log messages 12-31 to 12-32

displaying log queues 12-44, 12-47

displaying output definitions 12-8, 12-42

displaying reception of log messages 12-45

enabling 12-22, 12-46

enabling generation of log messages 12-23

enabling output definitions 12-8, 12-23

enabling reception of log messages 12-23

example configuration 12-9

fields in log message 12-3

filters 12-5, 12-12, 12-25, 12-40

format of log messages 12-3

forwarding to another router via SRLP 12-6, 12-25

forwarding to syslog server 12-7, 12-25

forwarding via SRLP 12-25

message filter comparison operators 12-9, 12-25

message filters 12-5, 12-8, 12-12, 12-25, 12-40

message ID 12-3

message reference 12-3 to 12-4

message severity 12-3, 12-5

message subtype 12-3 to 12-5

message text 12-3 to 12-4

message type 12-3 to 12-5

modifying log filters 12-9, 12-25

modifying output definitions 12-8, 12-25

module ID 12-3 to 12-4

Net Manage protocol 12-2, 12-5, 12-15, 12-20, 12-22 to 12-23, 12-29, 12-31, 12-45

output definitions 12-5, 12-8, 12-16, 12-19 to 12-21, 12-23 to 12-25, 12-40, 12-42

output to asynchronous port 12-6

output to RAM or NVS 12-6

PERMANENT output definition 12-6



processing of log messages 12-5
 purging 12-24
 queues 12-6, 12-24, 12-44, 12-47
 reception via Net Manage 12-15, 12-20,
 12-22 to 12-23, 12-29, 12-45
 reception via SRLP 12-15, 12-20, 12-22 to 12-23,
 12-29, 12-45
 secure router logging protocol (SRLP) 12-2, 12-4, 12-6,
 12-15, 12-22 to 12-23, 12-29, 12-45
 source file 12-3
 source line 12-3
 status 12-46
 syslog server 12-7
 TEMPORARY output definition 12-6
 Universal Coordinated Time (UTC) 12-4, 12-30
 logging in and out 1-10, 1-44 to 1-45
 LOGIN command 1-44
 LOGOFF command 1-45
 loopback plugs
 for testing 9-3
 lossless data compression 8-2
 lossy data compression 8-2
 LQM. See link quality management

M

Magic Number Option, LCP 3-5
 magic number, PPP 3-12
 managed devices, SNMP 16-2
 Management Information Base C-1 to C-16
 access mode 16-5
 address translation group, MIB-II MIB C-11
 Allied Telesyn Enterprise MIB C-3
 alliedTelesyn object C-3
 arBoardIndex object C-7
 arBoardMaxIndex object C-7
 arBoardTable object C-7 to C-9
 arInterfaceBoardIndex object C-7
 arInterfacePosition object C-7
 arInterfaces Group, Allied Telesyn Enterprise MIB C-6
 arInterfaces object C-4
 arInterfaceTable object C-7
 arSlotBoardIndex object C-7
 arSlotSlotIndex object C-7
 arSlotTable object C-7
 atIfIndex object C-12
 atNetAddress object C-12
 atPhysAddress object C-12
 atRouter object C-4
 atRouter subtree, Allied Telesyn Enterprise MIB C-4
 boards object C-5
 BRI Group, Allied Telesyn Enterprise MIB C-8
 bri object C-8
 briChanTable object C-8
 bridgeRouter object C-3 to C-4
 brIntTable object C-8
 brouterMib object C-4
 cc object C-8
 ccActiveCallTable object C-8
 ccAttachmentTable object C-8
 ccBchannelTable object C-8

ccCallLogTable object C-8
 ccCliListTable object C-8
 ccDetailsTable object C-8
 centreCOM-AR300LRouter object C-4
 centreCOM-AR300LURouter object C-4
 centreCOM-AR300Router object C-4
 centreCOM-AR300URouter object C-4
 centreCOM-AR310Router object C-4
 centreCOM-AR310URouter object C-4
 centreCOM-AR330Router object C-4
 centreCOM-AR350Router object C-4
 centreCOM-AR370Router object C-4
 centreCOM-AR370URouter object C-4
 centreCOM-AR390Router object C-4
 centreCOM-AR395Router object C-4
 centreCOM-AR720Router object C-4
 chip_68020_cpu object C-6
 chip_68302_cpu object C-6
 chip_68340_cpu object C-6
 chip_68360_cpu object C-6
 chip_860T_cpu object C-6
 chip_flash_1mb object C-6
 chip_flash_2mb object C-6
 chip_flash_3mb object C-6
 chip_flash_4mb object C-6
 chip_flash_6mb object C-6
 chip_flash_8mb object C-6
 chip_pem object C-6
 chip_ram_12mb object C-6
 chip_ram_16mb object C-6
 chip_ram_1mb object C-6
 chip_ram_20mb object C-6
 chip_ram_2mb object C-6
 chip_ram_32mb object C-6
 chip_ram_3mb object C-6
 chip_ram_4mb object C-6
 chip_ram_6mb object C-6
 chip_ram_8mb object C-6
 chip_rtc1 object C-6
 chip_rtc2 object C-6
 chip_rtc3 object C-6
 chip_rtc4 object C-6
 chips object C-6
 CMOT group, MIB-II MIB C-11
 combined notation 16-4
 configFile object C-9 to C-10
 createConfigFile object C-10
 deprecated status 16-6
 dot3 object C-13
 dotted notation 16-4
 EGP group, MIB-II MIB C-11
 egpNeighEventTrigger object C-12
 enterprise MIB C-3
 Ethernet Group, Allied Telesyn Enterprise MIB C-7
 ethernet object C-7
 Ethernet-like Collision Statistics group, Ethernet-like In-
 terface Types MIB C-13
 Ethernet-like Interface Types MIB C-13
 Ethernet-like Statistics group, Ethernet-like Interface
 Types MIB C-13



- ethIntTable* object C-7
- File Group, Allied Telesyn Enterprise MIB C-10
- file* object C-10
- fileTable* object C-10
- Firewall Group, Allied Telesyn Enterprise MIB C-10
- firewall* object C-10
- firewallTrap* trap C-10
- firewallTrapMessage* object C-10
- get-next PDU 16-6
- get-request PDU 16-6
- get-response PDU 16-6
- host* object C-14
- Host Resources Device group, Host Resources MIB C-14 to C-15
- Host Resources Installed Software group, Host Resources MIB C-14 to C-15
- Host Resources MIB C-14
- Host Resources Running Software group, Host Resources MIB C-14 to C-15
- Host Resources Running Software Performance group, Host Resources MIB C-14 to C-15
- Host Resources Storage group, Host Resources MIB C-14 to C-15
- Host Resources System group, Host Resources MIB C-14 to C-15
- hrDeviceDiskStorage* object C-15
- hrDeviceEntry* object C-15
- hrDeviceID* object C-4, C-15
- hrDeviceType* object C-15
- hrDiskStorageTable* object C-15
- hrFSTable* object C-15
- hrNetworkTable* object C-15
- hrPrinterTable* object C-15
- hrStorageAllocationUnits* object C-15
- hrSWRunTable* object C-14
- hrSystemDate* object C-15
- hrSystemInitialLoadDevice* object C-15
- hrSystemInitialLoadParameters* object C-15
- hrSystemMaxProcesses* object C-15
- hrSystemNumUsers* object C-15
- hrSystemProcesses* object C-15
- ICMP group, MIB-II MIB C-11
- iface_asyn* object C-5
- iface_bri* object C-5
- iface_eth* object C-5
- iface_pots* object C-5
- iface_pri* object C-5
- iface_syn* object C-5
- ifAdminStatus* object C-12
- ifIndex* object C-7 to C-9
- ifSpecific* object C-13
- ifType* object C-13
- ifTypes* object C-5
- Install Group, Allied Telesyn Enterprise MIB C-9
- install* object C-9
- Installed Software Group, Host Resources MIB C-16
- installHistoryTable* object C-9
- installTable* object C-9
- instance identifier 16-4
- interfaces group, MIB-II MIB C-11
- Internet-standard MIB 16-3
- IP group, MIB-II MIB C-11
- ipNetToMediaIffIndex* object C-12
- ipNetToMediaNetAddress* object C-12
- ipNetToMediaPhysAddress* object C-12
- ipNetToMediaType* object C-12
- ipRouteAge* object C-12
- ipRouteIffIndex* object C-12
- ipRouteMask* object C-12
- ipRouteMetric1* object C-12
- ipRouteMetric2* object C-12
- ipRouteMetric3* object C-12
- ipRouteMetric4* object C-12
- ipRouteMetric5* object C-12
- ipRouteNextHop* object C-12
- ipRouteType* object C-12
- ISDN Call Control Group, Allied Telesyn Enterprise MIB C-8
- licenceTable* object C-10
- Loader Group, Allied Telesyn Enterprise MIB C-9
- loader* object C-9
- loadStatus* object C-9
- loadTable* object C-9
- mandatory status 16-6
- mib-2 C-11
- MIB-II MIB C-11
- mibObject* object C-4
- Modules Group, Allied Telesyn Enterprise MIB C-7
- modules* object C-4
- not-accessible access mode 16-5
- object descriptor 16-4
- object identifier 16-4
- object instance 16-4
- object syntax 16-5
- Objects Group, Allied Telesyn Enterprise MIB C-4
- objects* object C-4
- obsolete status 16-6
- optional status 16-6
- ppr_a010* object C-5
- ppr_a011* object C-5
- ppr_a012* object C-5
- ppr_ar300* object C-5
- ppr_ar300L* object C-5
- ppr_ar300Lu* object C-5
- ppr_ar300u* object C-5
- ppr_ar310* object C-5
- ppr_ar310u* object C-5
- ppr_ar330* object C-5
- ppr_ar350* object C-5
- ppr_ar370* object C-5
- ppr_ar370u* object C-5
- ppr_ar390* object C-5
- ppr_ar395* object C-5
- ppr_ar720* object C-5
- ppr_icm_ar020* object C-5
- ppr_icm_ar021* object C-5
- ppr_icm_ar022* object C-5
- ppr_icm_ar023* object C-5
- ppr_icm_ar024* object C-5
- ppr_icm_ar025* object C-5

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

- PRI Group, Allied Telesyn Enterprise MIB C-9
- pri* object C-9
- priChanTable* object C-9
- priIntTable* object C-9
- products* object C-3
- Products subtree, Allied Telesyn Enterprise MIB C-3
- protocols* object C-4
- read-only access mode 16-5
- read-write access mode 16-5
- Running Software Group, Host Resources MIB C-15
- Running Software Performance Group, Host Resources MIB C-16
- set PDU 16-6
- SNMP group, MIB-II MIB C-12
- snmpEnableAuthenTraps* object C-12
- status 16-5
- sysinfo* object C-4
- system group, MIB-II MIB C-11
- TCP group, MIB-II MIB C-11
- tcpConnState* object C-12
- textual notation 16-4
- Transmission Group, MIB-II MIB C-13
- Transmission group, MIB-II MIB C-12
- trap PDU 16-6
- traps* object C-4
- UDP group, MIB-II MIB C-11
- write-only access mode 16-5
- Management Information Base (MIB)
 - Ethernet 2-23
 - Internet Protocol (IP) 6-23, 6-109, 16-1
 - Point-to-Point Protocol (PPP) 3-19
- management, using Telnet 1-12
- manager port, semipermanent 1-4, 1-12, 1-53, 1-77
- MANAGER privilege 1-4
- mandatory status 16-6
- Maximum Receive Unit Option, LCP 3-5
- Maximum Received Reconstructed Unit Option, LCP 3-5
- memory
 - displaying
 - contents 1-14, 1-35
 - free 1-13, 1-59
 - modifying
 - contents 1-14, 1-45
- message format, SNMP protocol 16-6
- metrics
 - boosts 6-16
 - Internet Protocol (IP) 6-16
- mib-2 C-11
- mibObject* MIB object C-4
- MIOX 5-18, 5-23 to 5-24
- MIOX circuits 5-4, 5-9 to 5-10, 5-14 to 5-17, 5-19, 5-26
- mixed mode, Time Division Multiplexing (TDM) 11-2
- MODIFY command 1-45
- modifying
 - cadences, PBX 14-20
 - country, PBX 14-20
 - domain name 4-80
 - extensions, PBX 14-21
 - firewall rules 17-4, 17-24
 - groups, PBX 14-24

- interfaces, PPP 3-49
- passwords 1-54
- PPP interfaces 3-49
- PPP templates 3-54
- prefixes, PBX 14-20
- route templates 6-101
- templates 3-54
- territory, PBX 14-20
- Modules Group, Allied Telesyn Enterprise MIB C-7
- modules* MIB object C-4
- monitoring
 - firewall 17-6 to 17-8
- monitoring operation 1-13
- multicast addresses 2-6, 2-29
- multicasting, IP 6-31
- multihoming, Internet Protocol (IP) 6-9
- multilink, PPP 3-6

N

- name
 - displaying 1-78
 - setting 1-55
- name server 6-95, 6-102
- name server, Internet Protocol (IP) 6-95
- Names, MIB objects 16-4
- naming files 1-15
 - using wildcards 1-16
- naming interfaces 2-2
- NAT
 - used with firewall 17-6
- Net Manage protocol 12-2, 12-5
- network classes 6-6
- Network Control Protocol (NCP) 3-3 to 3-4
- network management protocol 16-2
- Network Management Station (NMS) 16-2
- non-associated signalling, ISDN 4-15
- not-accessible, MIB object 16-5
- notation, MIB object identifiers
 - combined 16-4
 - dotted 16-4
 - textual 16-4

O

- object descriptor 16-4
- object identifier 16-4
 - combined notation 16-4
 - dotted notation 16-4
 - textual notation 16-4
- object instance 16-4
- object syntax 16-5
- object, MIB
 - access mode 16-5
 - deprecated 16-6
 - mandatory 16-6
 - not-accessible 16-5
 - obsolete 16-6
 - optional 16-6
 - read-only 16-5
 - read-write 16-5
 - status 16-5



- write-only 16-5
- Objects Group, Allied Telesyn Enterprise MIB C-4
- objects MIB object C-4
- obsolete status 16-6
- online help 1-41
 - setting system help file 1-50
- Operation 1-1 to 1-83
 - adding
 - users 1-28
 - administrator name
 - displaying 1-78
 - setting 1-54
 - asynchronous port security 1-11
 - boot script 1-6
 - booting the router 1-13, 1-49
 - changing
 - passwords 1-54
 - command
 - editing 1-4
 - history 1-4
 - processor 1-3
 - commands
 - ACTIVATE FLASH COMPACTION 1-26
 - ADD ALIAS 1-28
 - ADD USER 1-28
 - CREATE CONFIG 1-30
 - CREATE FFILE 1-31
 - DELETE ALIAS 1-31
 - DELETE FFILE 1-32
 - DELETE FILE 1-32
 - DELETE INSTALL 1-33
 - DELETE USER 1-33
 - DISABLE HTTP SERVER 1-34
 - DISABLE USER 1-35
 - DUMP 1-35
 - EDIT 1-37
 - ENABLE HTTP DEBUG 1-40
 - ENABLE HTTP SERVER 1-40
 - ENABLE USER 1-41
 - HELP 1-41
 - LOAD 1-42
 - LOGIN 1-44
 - LOGOFF 1-45
 - MODIFY 1-45
 - PURGE USER 1-46
 - RENAME 1-46
 - RESET HTTP SERVER 1-47
 - RESET LOADER 1-47
 - RESET USER 1-48
 - RESTART 1-49
 - SET CONFIG 1-49
 - SET HELP 1-50
 - SET INSTALL 1-50
 - SET LOADER 1-51
 - SET MANAGER PORT 1-53
 - SET PASSWORD 1-54
 - SET SYSTEM CONTACT 1-54
 - SET SYSTEM LOCATION 1-55
 - SET SYSTEM NAME 1-55
 - SET SYSTEM TERRITORY 1-56
 - SET TIME 1-56
 - SET USER 1-57
 - SHOW ALIAS 1-58
 - SHOW BUFFER 1-59
 - SHOW CONFIG 1-61
 - SHOW CPU 1-63
 - SHOW DEBUG 1-63
 - SHOW EXCEPTION 1-64
 - SHOW FFILE 1-65, 1-67
 - SHOW FLASH 1-67
 - SHOW FLASH PHYSICAL 1-69
 - SHOW HTTP CLIENT 1-69
 - SHOW HTTP DEBUG 1-70
 - SHOW HTTP SERVER 1-72
 - SHOW HTTP SESSION 1-71
 - SHOW INSTALL 1-74
 - SHOW LOADER 1-75
 - SHOW MANAGER PORT 1-77
 - SHOW STARTUP 1-77
 - SHOW SYSTEM 1-78
 - SHOW TIME 1-80
 - SHOW USER 1-80
 - UPLOAD 1-82
 - configuration
 - displaying dynamic 1-61
 - saving dynamic 1-30
 - setting default 1-49
 - contact name
 - displaying 1-78
 - setting 1-54
 - counters 1-11
 - CPU utilisation 1-13, 1-63
 - creating
 - files 1-16
 - FLASH files 1-31
 - debugging 1-13, 1-63
 - default configuration, setting 1-49
 - default install 1-23
 - deleting
 - files 1-16, 1-32
 - FLASH files 1-32
 - install information 1-33
 - users 1-33
 - disabling
 - users 1-35
 - displaying
 - administrator name 1-78
 - clock 1-80
 - contact name 1-78
 - date 1-80
 - directory of files 1-16
 - dynamic configuration 1-61
 - exceptions 1-13, 1-64
 - file downloads 1-75
 - files 1-67
 - FLASH files 1-67
 - FLASH memory 1-15, 1-67, 1-69
 - free memory 1-13, 1-59
 - help 1-5, 1-41
 - install information 1-74



- location 1-78
- memory contents 1-14, 1-35
- name 1-78
- semipermanent manager port 1-77
- startup status 1-13, 1-77
- system clock 1-80
- system date 1-80
- system information 1-78
- system location 1-78
- system name 1-78
- system territory 1-78
- system time 1-80
- territory 1-78
- time 1-80
- users 1-80
- downloading files 1-16, 1-42, 1-47, 1-51, 1-75
- dynamic configuration
 - displaying 1-61
 - saving 1-30
- editing files 1-16, 1-37
- editor 1-18, 1-37
- enabling
 - users 1-41
- entering commands 1-3 to 1-4
- event logging 1-13
- exceptions, displaying 1-13, 1-64
- fatal errors 1-13, 1-64
- fault diagnosis 1-13
- file naming conventions 1-15
- file subsystem 1-15
- file types 1-15
- files
 - creating 1-16
 - deleting 1-16, 1-32
 - displaying 1-67
 - displaying directory 1-16
 - downloading 1-16, 1-42, 1-47, 1-51, 1-75
 - editing 1-16, 1-37
 - renaming 1-46
 - uploading 1-82
- FLASH files 1-16
 - creating 1-31
 - deleting 1-32
 - displaying 1-65, 1-67
 - displaying directory 1-17
 - downloading 1-42, 1-47, 1-51, 1-75
 - editing 1-37
 - renaming 1-46
 - uploading 1-82
- FLASH memory 1-14 to 1-15
 - compaction 1-17, 1-26
 - displaying 1-67, 1-69
- help
 - displaying 1-41
 - setting system help file 1-50
- help file 1-5
- HTTP server 1-19 to 1-20
 - disabling 1-34
 - displaying client status 1-69
 - displaying debugging status 1-70
 - displaying session information 1-71
 - displaying status 1-72
 - enabling 1-40
 - enabling debugging 1-40
 - resetting 1-47
 - resolving URLs 1-20
- install information 1-23
 - default install 1-23
 - deleting 1-33
 - displaying 1-74
 - preferred install 1-23
 - setting 1-50
 - temporary install 1-23
- location
 - displaying 1-78
 - setting 1-55
- logging events 1-13
- logging in and out 1-10, 1-44 to 1-45
- manager port, semipermanent 1-4, 1-12, 1-53, 1-77
- MANAGER privilege 1-4
- memory
 - displaying 1-14
 - displaying contents 1-35
 - displaying FLASH 1-15
 - displaying free 1-13, 1-59
 - FLASH 1-14 to 1-15, 1-67, 1-69
 - FLASH compaction 1-17, 1-26
 - modifying 1-14, 1-45
- modifying
 - memory 1-14, 1-45
 - passwords 1-54
 - users 1-57
- monitoring 1-13
- name
 - displaying 1-78
 - setting 1-55
- online help 1-5, 1-41
- passwords 1-9
 - changing 1-54
 - recovering 1-11
- patches 1-20
 - downloading 1-22, 1-42, 1-47, 1-51, 1-75
- preferred install 1-23
- privilege levels 1-3
- prompt 1-3
- purging
 - users 1-46
- RADIUS servers 1-7
- recovering passwords 1-11
- remote management 1-12
- renaming
 - files 1-46
 - FLASH files 1-46
- resetting
 - users 1-48
- restarting the router 1-13, 1-49
- retrieving configurations 1-6
- saving default configuration 1-49
- saving dynamic configuration 1-6, 1-30
- scripts, boot 1-6



- self tests 1-21
 - semipermanent manager port 1-4, 1-12, 1-53, 1-77
 - setting
 - administrator name 1-54
 - clock 1-56
 - contact name 1-54
 - date 1-56
 - install information 1-50
 - location 1-55
 - name 1-55
 - semipermanent manager port 1-53
 - system clock 1-56
 - system date 1-56
 - system help file 1-50
 - system information 1-54 to 1-56
 - system location 1-55
 - system name 1-55
 - system territory 1-56
 - system time 1-56
 - territory 1-56
 - time 1-56
 - snapshot 1-13, 1-63
 - software releases 1-20
 - deleting 1-33
 - displaying 1-74
 - downloading 1-22, 1-42, 1-47, 1-51, 1-75
 - examples 1-24
 - installing a compressed release 1-25
 - installing a standard release 1-24
 - naming 1-20
 - setting 1-50
 - startup operations 1-21
 - startup status 1-13, 1-77
 - system information
 - displaying 1-78
 - setting 1-54 to 1-56
 - system location
 - displaying 1-78
 - setting 1-55
 - system name
 - displaying 1-78
 - setting 1-55
 - system territory
 - displaying 1-78
 - setting 1-56
 - TACACS servers 1-7
 - Telnet 1-12
 - Telnet from the router 1-11
 - temporary install 1-23
 - territory
 - displaying 1-78
 - setting 1-56
 - text editor 1-18, 1-37
 - uploading files 1-82
 - User Authentication Database 1-7 to 1-8
 - adding users 1-8, 1-28
 - counters 1-11
 - deleting users 1-8, 1-33
 - disabling users 1-35
 - displaying users 1-9, 1-80
 - enabling users 1-41
 - logging in and out 1-10, 1-44 to 1-45
 - modifying users 1-8, 1-57
 - passwords 1-9
 - purging users 1-46
 - recovering passwords 1-11
 - resetting users 1-48
 - security 1-9
 - User Authentication Facility 1-7
 - asynchronous port security 1-11
 - semipermanent manager port 1-12
 - Telnetting from the router 1-11
 - USER privilege 1-3
 - users
 - adding 1-28
 - deleting 1-33
 - disabling 1-35
 - displaying 1-80
 - enabling 1-41
 - modifying 1-57
 - purging 1-46
 - resetting 1-48
 - wildcards, in file names 1-16
 - operation
 - commands
 - SET SYSTEM TERRITORY 4-16, 4-33, 4-35
 - optional status 16-6
- ## P
- PABX. See Telephony Services
 - packet assembler/disassembler (PAD) 5-2
 - PAD, X.25 5-2
 - PAP. See Password Authentication Protocol (PAP)
 - parameters, in scripts 13-4
 - Password Authentication Protocol (PAP) 3-12
 - passwords 1-9
 - changing 1-54
 - recovering 1-11
 - patches 1-20
 - downloading 1-22, 1-42, 1-47, 1-51, 1-75, 1-82
 - payload compression 8-3
 - PBX. See Telephony Services
 - periodic triggers 10-2
 - permanent virtual circuits (PVC) 5-2
 - PING 6-25, 6-82, 6-103, 6-137, 6-145
 - PING command 6-82
 - PING utility 6-36
 - Point-to-Point Protocol (PPP) 3-1
 - adding
 - physical links 3-28
 - aggregating links 3-6, 3-22
 - Always On/Demand ISDN (AODI) 3-19
 - authentication 3-12 to 3-15, 3-28
 - BACP. See Bandwidth Allocation Control Protocol
 - Bandwidth Allocation Control Protocol (BACP) 3-7
 - Bandwidth Allocation Protocol (BAP) 3-7
 - bandwidth on demand 3-8, 3-25
 - BAP. See Bandwidth Allocation Protocol
 - callback 3-11
 - Challenge-Handshake Authentication Protocol (CHAP)



- 3-13
- channel aggregation 3-6, 3-22
- commands
 - ACTIVATE PPP 3-27
 - ADD PPP 3-28
 - CREATE PPP 3-31
 - CREATE PPP TEMPLATE 3-36
 - DELETE PPP 3-41
 - DESTROY PPP 3-41
 - DESTROY PPP TEMPLATE 3-42
 - DISABLE PPP 3-42
 - DISABLE PPP DEBUG 3-43
 - DISABLE PPP TEMPLATE DEBUG 3-43
 - ENABLE PPP 3-44
 - ENABLE PPP DEBUG 3-45
 - ENABLE PPP TEMPLATE DEBUG 3-46
 - PURGE PPP 3-47
 - RESET PPP 3-48
 - SET PPP 3-49
 - SET PPP TEMPLATE 3-54
 - SHOW PPP 3-58
 - SHOW PPP CONFIG 3-59
 - SHOW PPP COUNT 3-65
 - SHOW PPP DEBUG 3-75
 - SHOW PPP IDLETIMER 3-76
 - SHOW PPP LIMITS 3-77
 - SHOW PPP MULTILINK 3-78
 - SHOW PPP NAMESERVER 3-80
 - SHOW PPP TEMPLATE 3-80
 - SHOW PPP TXSTATUS 3-84
- compression 3-25, 3-28
- compression, Van Jacobson's 3-18
- configuration
 - displaying 3-59
- configuration examples 3-20
- control protocols 3-3 to 3-4
- counters 3-19
 - displaying 3-65
 - displaying activity 3-17
 - resetting activity 3-17, 3-48
- creating
 - interfaces 3-31
 - templates 3-36
- debugging 3-17, 3-43, 3-45, 3-75
 - templates 3-43, 3-46
- deleting
 - physical links 3-41
- destroying
 - interfaces 3-41
 - templates 3-42
- dial-on-demand 3-8, 3-23
- disabling
 - debugging, interfaces 3-43
 - debugging, templates 3-43
 - interfaces 3-42
- displaying
 - activity counters 3-17
 - configuration 3-59
 - counters 3-65
 - debugging 3-75
 - interfaces 3-58
 - multilink information 3-78
 - nameservers 3-80
 - templates 3-80
 - timers 3-76
 - transmission queue 3-84
- dynamic interfaces
 - templates 3-9
- enabling
 - debugging, interfaces 3-45
 - debugging, templates 3-46
 - interfaces 3-44
- encapsulation 3-3
- encryption 3-25
- events 3-4
- example configurations 3-20
- interfaces
 - adding physical links 3-28
 - creating 3-31
 - deleting physical links 3-41
 - destroying 3-41
 - disabling 3-42
 - displaying 3-58
 - enabling 3-44
 - modifying 3-49
 - purging 3-47
 - resetting 3-48
- limiting connection time and data throughput 3-16, 3-31, 3-36, 3-48 to 3-49, 3-54
- Link Control Protocol (LCP) 3-4, 3-19
 - Authentication Protocol Option 3-5
 - Endpoint Discriminator Option 3-5
 - Link Discriminator Option 3-5 to 3-6
 - Link Quality Reporting (LQR) Option 3-5 to 3-6
 - Magic Number Option 3-5
 - Maximum Receive Unit Option 3-5
 - Maximum Received Reconstructed Unit Option 3-5
 - options 3-5
- link management 3-16, 3-31, 3-36, 3-48 to 3-49, 3-54
- link quality management 3-19, 3-24, 3-28
- Link Quality Reporting (LQR) 3-6
- magic number 3-12
- MIB 3-19
- modifying
 - interfaces 3-49
- multilink 3-6
 - displaying 3-78
- nameservers
 - displaying 3-80
 - setting 3-49
- NCP 3-4
- Network Control Protocol (NCP) 3-4
- options 3-17
- Password Authentication Protocol (PAP) 3-12
- protocols 3-3
- purging configuration 3-47
- resetting interfaces 3-48
- resetting
 - activity counters 3-17, 3-48
- setting



- activity limits 3-16 to 3-17, 3-31, 3-36
- standards 3-3
- states 3-4
- support for 3-17
- templates 3-9
 - creating 3-36
 - debugging 3-43, 3-46
 - destroying 3-42
 - displaying 3-80
 - modifying 3-54
- thresholds for connection time and data throughput 3-16, 3-31, 3-36, 3-48 to 3-49, 3-54
- timers 3-19, 3-76
- transmission queue 3-84
- Van Jacobson 3-17
- Van Jacobson's compression 3-18
- policies, firewall 17-3 to 17-4
 - adding interfaces 17-4, 17-11
 - creating 17-3, 17-16
 - deleting interfaces 17-4, 17-17
 - destroying 17-3, 17-20
 - displaying 17-4 to 17-5, 17-26
- polling, SNMP protocol 16-8
- ppr_ar010* MIB object C-5
- ppr_ar011* MIB object C-5
- ppr_ar012* MIB object C-5
- ppr_ar300* MIB object C-5
- ppr_ar300L* MIB object C-5
- ppr_ar300Lu* MIB object C-5
- ppr_ar300u* MIB object C-5
- ppr_ar310* MIB object C-5
- ppr_ar310u* MIB object C-5
- ppr_ar330* MIB object C-5
- ppr_ar350* MIB object C-5
- ppr_ar370* MIB object C-5
- ppr_ar370u* MIB object C-5
- ppr_ar390* MIB object C-5
- ppr_ar395* MIB object C-5
- ppr_ar720* MIB object C-5
- ppr_icm_ar020* MIB object C-5
- ppr_icm_ar021* MIB object C-5
- ppr_icm_ar022* MIB object C-5
- ppr_icm_ar023* MIB object C-5
- ppr_icm_ar024* MIB object C-5
- ppr_icm_ar025* MIB object C-5
- Predictor compression 8-7
- PRI Group, Allied Telesyn Enterprise MIB C-9
- pri* MIB object C-9
- priChanTable* MIB object C-9
- priIntTable* MIB object C-9
- primary name server 6-95
- privilege levels 1-3
 - MANAGER 1-4
 - USER 1-3
- products* MIB object C-3
- Products sub-tree, Allied Telesyn Enterprise MIB C-3
- protocols
 - RIP 6-14
- protocols* MIB object C-4
- proxy ARP 6-10

- purge
 - Internet Protocol (IP) configuration 6-84
- PURGE BOOTP RELAY command 6-84
- PURGE IP command 6-84
- PURGE LOG command 12-24
- PURGE PORT command 2-15
- PURGE PPP command 3-47
- PURGE TDM command 11-6
- PURGE TRIGGER command 10-13
- PURGE USER command 1-46
- purging
 - PPP configuration 3-47
 - users 1-46
- PVC. See X.25

Q

- Q.931 4-15

R

- RADIUS server
 - adding to rule 17-14
- read-only, MIB object 16-5
- read-write, MIB object 16-5
- reboot
 - triggers 10-2
- reinitialise
 - Internet Protocol (IP) module 6-84
- Remote CAPI. See Common Application Programmer's Interface (CAPI)
- remote management 1-12
- RENAME command 1-46
- renaming
 - files 1-46
 - FLASH files 1-46
- reset
 - Ethernet 2-16
- RESET BRI command 4-72
- RESET BRI COUNTERS command 4-72
- RESET ENCO COUNTERS command 8-7
- RESET ETH command 2-16
- RESET ETH COUNTERS command 2-16
- RESET HTTP SERVER command 1-47
- RESET IP command 6-84
- RESET IP COUNTER command 6-85
- RESET IP INTERFACE command 6-85
- RESET LOADER command 1-47
- RESET PORT command 2-17
- RESET PORT COUNTERS command 2-17
- RESET PORT HISTORY command 2-18
- RESET PPP command 3-48
- RESET Q931 command 4-73
- RESET TEST INTERFACE command 9-10
- RESET USER command 1-48
- RESET X25T command 5-18
- resetting
 - hardware tests 9-10
 - interfaces, PPP 3-48
 - PPP interfaces 3-48
 - tests 9-10
- resetting



- activity counters on PPP interfaces 3-17, 3-48
- counters
 - ENCO 8-7
- restart
 - Internet Protocol (IP) module 6-84
- RESTART command 1-49
- restarting the router 1-13, 1-49
- route templates
 - adding 6-60
 - deleting 6-69
 - displaying 6-135
 - modifying 6-101
- route templates, IP 6-19
- router type
 - Internet Protocol (IP) 6-72, 6-74, 6-77, 6-79
- routes
 - adding Internet Protocol (IP) 6-13, 6-57
 - changing Internet Protocol (IP) 6-98
 - deleting Internet Protocol (IP) 6-13, 6-68
 - disabling Internet Protocol (IP) 6-76
 - displaying Internet Protocol (IP) 6-13, 6-35, 6-130
 - enabling Internet Protocol (IP) 6-81
 - Internet Protocol (IP) 6-12, 6-57, 6-68, 6-76, 6-81, 6-98
 - modifying Internet Protocol (IP) 6-98
 - setting Internet Protocol (IP) 6-98
- routing
 - information filters 6-13
 - information filters, Internet Protocol (IP) 6-59, 6-69, 6-100, 6-134
 - Internet Protocol (IP) 6-12
 - protocols
 - Routing Information Protocol (RIP) 6-14
- Routing Information Protocol (RIP) 6-14, 6-56, 6-67, 6-96 to 6-97, 6-126 to 6-128
- routing table, Internet Protocol (IP) 6-12, 6-130
- rules, firewall 17-4 to 17-5
 - adding 17-4, 17-14
 - deleting 17-4, 17-19
 - displaying 17-5, 17-26
 - modifying 17-4, 17-24
- Running Software Group, Host Resources MIB C-15
- Running Software Performance Group, Host Resources MIB C-16

S

- S/T interfaces, Basic Rate Access 4-4
- saving dynamic configuration 1-30
- scripting. *See* scripts
- scripts 13-1 to 13-12
 - activating by trigger 13-2
 - activating from command line 13-2, 13-5
 - adding 13-6
 - BOOT.CFG 13-2
 - calling other scripts 13-4
 - CFG file type 13-2
 - commands
 - ACTIVATE SCRIPT 13-5
 - ADD SCRIPT 13-2, 13-6
 - DEACTIVATE SCRIPT 13-7
 - DELETE SCRIPT 13-8

- SET SCRIPT 13-2, 13-10
- SHOW SCRIPT 13-2, 13-11
- WAIT 13-8, 13-12
- control structures 13-4, 13-8
- creating 13-2
- creating using script commands 13-2, 13-6
- creating using text editor 13-3
- deactivating 13-7
- deleting 13-8
- displaying 13-2, 13-11
- file format 13-2
- halting 13-7
- if..then..else..endif 13-4, 13-8
- loading from asynchronous port 13-3
- loading from TFTP server 13-3
- modifying 13-10
- modifying using script commands 13-2, 13-8, 13-10
- output to logging facility 13-4
- output to terminal 13-4
- parameters 13-4
- pausing 13-12
- playing 13-2
- redirecting output 13-4
- SCP file type 13-2
- script commands 13-2
- using scripts 13-4
- scripts, boot 1-6
- secondary name server 6-102
- secure router logging protocol (SRLP) 12-4, 12-6
- security
 - asynchronous ports 1-11
 - options for Internet Protocol (IP) 6-26
- security policies, firewall 17-3 to 17-4
- self tests 1-21
- semipermanent manager port 1-4, 1-12, 1-53, 1-77
- SET BOOTP MAXHOPS command 6-86
- SET BRI command 4-73
- SET CONFIG command 1-49
- SET DHCP POLICY command 15-17
- SET DTEADDRESS command 5-8
- SET ENCO SW command 8-7
- SET FIREWALL POLICY RULE command 17-24
- SET HELP command 1-50
- SET INSTALL command 1-50
- SET INTERFACE TRAPLIMIT command 2-18
- SET IP ARP command 6-86
- SET IP AUTONOMOUS command 6-87
- SET IP FILTER command 6-88
- SET IP HOST command 6-91
- SET IP INTERFACE command 6-92
- SET IP LOCAL command 6-94
- SET IP NAMESERVER command 6-95
- SET IP RIP 6-96
- SET IP RIPTIMER command 6-97
- SET IP ROUTE command 6-98
- SET IP ROUTE FILTER command 6-100
- SET IP ROUTE TEMPLATE command 6-101
- SET IP SECONDARYNAMESERVER command 6-102
- SET ISDN CALL command 4-75
- SET ISDN DOMAINNAME command 4-80



- SET ISDN LOG command 4-81
- SET LOADER command 1-51
- SET LOG OUTPUT command 12-25
- SET LOG RECEIVE command 12-29
- SET LOG UTCOFFSET command 12-30
- SET MANAGER PORT command 1-53
- SET MIOX CIRCUIT command 5-19
- SET MIOX command 5-18
- SET PASSWORD command 1-54
- SET PBX command 14-20
- SET PBX EXTENSION command 14-21
- SET PBX GROUP command 14-24
- set PDU 16-6
- SET PING command 6-103
- SET PORT command 2-19
- SET PPP command 3-49
- SET PPP TEMPLATE command 3-54
- SET Q931 command 4-84
- SET SCRIPT command 13-2, 13-10
- SET SNMP COMMUNITY command 16-19
- SET SYSTEM CONTACT command 1-54
- SET SYSTEM LOCATION command 1-55
- SET SYSTEM NAME command 1-55
- SET SYSTEM TERRITORY command 1-56, 4-16, 4-33, 4-35
- SET TELNET command 7-6
- SET TIME command 1-56
- SET TRACE command 6-104
- SET TRIGGER command 10-13
- SET TTY command 7-7
- SET USER command 1-57
- SET X25T command 5-21
- SET X25T CPAR command 5-22
- setting
 - activity limits on PPP interfaces 3-16 to 3-17, 3-31, 3-36
 - clock 1-56
 - date 1-56
 - default configuration 1-49
 - install information 1-50
 - semipermanent manager port 1-53
 - system clock 1-56
 - system date 1-56
 - system help file 1-50
 - system information 1-56
 - system time 1-56
 - time 1-56
- SHOW ALIAS command 1-58
- SHOW BOOTP RELAY command 6-105
- SHOW BRI CONFIGURATION command 4-86
- SHOW BRI COUNTERS command 4-88
- SHOW BRI CTEST command 4-94
- SHOW BRI DEBUG command 4-95
- SHOW BRI STATE command 4-96
- SHOW BRI TEST command 4-100
- SHOW BUFFER command 1-59
- SHOW CONFIG command 1-61
- SHOW CPU command 1-63
- SHOW DEBUG command 1-63
- SHOW DHCP CLIENT command 15-23
- SHOW DHCP command 15-22
- SHOW DHCP POLICY command 15-24
- SHOW DHCP RANGE command 15-25
- SHOW ENCO CHANNEL command 8-9
- SHOW ENCO command 8-8
- SHOW ENCO COUNTERS command 8-13
- SHOW ETH CONFIGURATION command 2-22
- SHOW ETH COUNTERS command 2-23
- SHOW ETH MACADDRESS command 2-29
- SHOW ETH RECEIVE command 2-29
- SHOW EXCEPTION command 1-64
- SHOW FFILE command 1-65, 1-67
- SHOW FIREWALL command 17-25
- SHOW FIREWALL POLICY command 17-26
- SHOW FIREWALL POLICY SESSION command 17-32
- SHOW FLASH command 1-67
- SHOW FLASH PHYSICAL command 1-69
- SHOW HTTP CLIENT command 1-69
- SHOW HTTP DEBUG command 1-70
- SHOW HTTP SERVER command 1-72
- SHOW HTTP SESSION command 1-71
- SHOW INSTALL command 1-74
- SHOW INTERFACE command 2-30
- SHOW IP ARP command 6-108
- SHOW IP command 6-106
- SHOW IP COUNTER command 6-109
- SHOW IP DEBUG command 6-116
- SHOW IP FILTER command 6-117
- SHOW IP HELPER command 6-119
- SHOW IP HOST command 6-120
- SHOW IP INTERFACE command 6-121
- SHOW IP POOL command 6-124
- SHOW IP RIP command 6-126
- SHOW IP RIP COUNTERS command 6-128
- SHOW IP RIPTIMER command 6-127
- SHOW IP ROUTE command 6-130
- SHOW IP ROUTE FILTER command 6-134
- SHOW IP ROUTE TEMPLATE command 6-135
- SHOW IP TRUSTED command 6-136
- SHOW IP UDP command 6-136
- SHOW ISDN CALL command 4-103
- SHOW ISDN CLILIST command 4-107
- SHOW ISDN DOMAINNAME command 4-108
- SHOW ISDN LOG command 4-109
- SHOW LAPD command 4-110
- SHOW LAPD COUNT command 4-112
- SHOW LAPD STATE command 4-114
- SHOW LOADER 1-75
- SHOW LOADER command 1-75
- SHOW LOG command 12-31 to 12-32
- SHOW LOG COUNTERS command 12-38
- SHOW LOG OUTPUT command 12-40, 12-42
- SHOW LOG QUEUE command 12-44, 12-47
- SHOW LOG RECEIVE command 12-45
- SHOW LOG STATUS command 12-46
- SHOW MANAGER PORT command 1-77
- SHOW MIOX CIRCUIT command 5-26
- SHOW MIOX command 5-23
- SHOW MIOX COUNT command 5-24
- SHOW PBX CALL command 14-26
- SHOW PBX command 14-25
- SHOW PBX EXTENSION command 14-27



- SHOW PBX GROUP command 14-29
- SHOW PING command 6-137
- SHOW PORT command 2-33
- SHOW PPP command 3-58
- SHOW PPP CONFIG command 3-59
- SHOW PPP COUNT command 3-65
- SHOW PPP DEBUG command 3-75
- SHOW PPP IDLETIMER command 3-76
- SHOW PPP LIMITS command 3-77
- SHOW PPP MULTILINK command 3-78
- SHOW PPP NAMESERVER command 3-80
- SHOW PPP TEMPLATE command 3-80
- SHOW PPP TXSTATUS command 3-84
- SHOW Q931 command 4-114
- SHOW Q931 SPID command 4-117
- SHOW SCRIPT command 13-2, 13-11
- SHOW SNMP command 16-20
- SHOW SNMP COMMUNITY command 16-22
- SHOW STARTUP command 1-77
- SHOW SYSTEM command 1-78
- SHOW TCP command 6-139
- SHOW TDM command 11-6
- SHOW TEST command 9-10
- SHOW TIME command 1-80
- SHOW TRACE command 6-143
- SHOW TRIGGER command 10-16
- SHOW TTY command 7-8
- SHOW USER command 1-80
- SHOW X25T command 5-30
- SHOW X25T CPAR command 5-34
- Simple Network Management Protocol (SNMP) 6-23, 16-1 to 16-23
 - access mode 16-5
 - agent
 - disabling 16-16
 - displaying 16-20
 - enabling 16-17
 - authentication 16-10
 - authentication trap
 - disabling 16-16
 - enabling 16-18
 - commands
 - ADD SNMP COMMUNITY 16-13
 - CREATE SNMP COMMUNITY 16-14
 - DELETE SNMP COMMUNITY 16-15
 - DESTROY SNMP COMMUNITY 16-16
 - DISABLE SNMP 16-16
 - DISABLE SNMP AUTHENTICATE_TRAP 16-16
 - DISABLE SNMP COMMUNITY 16-17
 - ENABLE SNMP 16-17
 - ENABLE SNMP AUTHENTICATE_TRAP 16-18
 - ENABLE SNMP COMMUNITY 16-18
 - SET SNMP COMMUNITY 16-19
 - SHOW SNMP 16-20
 - SHOW SNMP COMMUNITY 16-22
 - communities 16-9
 - adding 16-13
 - creating 16-14
 - deleting 16-15
 - destroying 16-16
 - disabling 16-17
 - displaying 16-22
 - enabling 16-18
 - modifying 16-19
 - community 16-8
 - configuration example 16-11
 - deprecated status 16-6
 - example configuration 16-11
 - get-next PDU 16-6
 - get-request PDU 16-6
 - get-response PDU 16-6
 - instance 16-4
 - instance identifier 16-4
 - interface link traps 2-12 to 2-14, 2-18
 - Internet-standard MIB 16-3
 - Internet-standard Network Management Framework 16-2
 - link traps 2-12 to 2-14, 2-18
 - managed devices 16-2
 - Management Information Base (MIB) 16-2 to 16-3
 - managing interfaces 2-12
 - mandatory status 16-6
 - message format 16-6
 - names, object 16-4
 - network management protocol 16-2
 - Network Management Station (NMS) 16-2
 - not-accessible access mode 16-5
 - object descriptor 16-4
 - object identifier 16-4
 - object names 16-4
 - object syntax 16-5
 - obsolete status 16-6
 - optional status 16-6
 - polling 16-8
 - read-only access mode 16-5
 - read-write access mode 16-5
 - set PDU 16-6
 - status 16-5
 - structure of information 16-2 to 16-3
 - syntax 16-5
 - trap 16-7
 - trap PDU 16-6
 - view 16-8
 - write-only access mode 16-5
- slotted interfaces 4-28
- SNAP
 - encapsulation on Ethernet 2-4
- SNMP group, MIB-II MIB C-12
- snmpEnableAuthenTraps* MIB object C-12
- software releases 1-20
 - deleting 1-33
 - displaying 1-74
 - downloading 1-22, 1-42, 1-47, 1-51, 1-75, 1-82
 - examples 1-24
 - installing a compressed release 1-25
 - installing a standard release 1-24
 - naming 1-20
 - setting 1-50
- SPID. See Integrated Services Digital Network (ISDN)
- SRLP. See Logging Facility:secure router logging protocol



- (SRLP)
- STAC LZS compression 8-7
- standards
 - Point-to-Point Protocol (PPP) 3-3
- startup operations 1-21
- startup status 1-13, 1-77
- stateful inspection, firewall 17-2
- states
 - Point-to-Point Protocol (PPP) 3-4
- statistics, compression 8-5 to 8-6
- status, MIB object 16-5
- STOP PING command 6-145
- STOP TRACE command 6-145
- Structure of Management Information (SMI) 16-2 to 16-3
- subnet 6-8
- subnet mask 6-8
- support
 - for ISDN 4-6
- SVC. See X.25
- switched virtual circuits (SVC) 5-2
- syntax, MIB object 16-5
- sysinfo MIB object C-4
- system clock
 - displaying 1-80
 - setting 1-56
- system date
 - displaying 1-80
 - setting 1-56
- system group, MIB-II MIB C-11
- system information
 - displaying 1-78, 1-80
 - setting 1-54 to 1-56
- system time
 - displaying 1-80
 - setting 1-56

T

- Tables, MIB
 - Board C-7
 - Boards C-7
 - BRI Channel C-8
 - BRI Interface C-8
 - Disk Storage C-15
 - Ethernet Interfaces C-7
 - File C-10
 - File System C-15
 - Host Resources Running Software C-14
 - Install C-9
 - Install History C-9
 - Interface C-7
 - ISDN Active Call C-8
 - ISDN B Channel C-8
 - ISDN Call Attachment C-8
 - ISDN Call Details C-8
 - ISDN Call Log C-8
 - ISDN CLI List C-8
 - Load C-9
 - Network Devices C-15
 - PRI Channel C-9
 - PRI Interface C-9

- Printer C-15
- Release Licence C-10
- Slot C-7
- TCP group, MIB-II MIB C-11
- tcpConnState MIB object C-12
- TDM. See Time Division Multiplexing (TDM)
- Telephony Services 14-1
 - autodial 14-13, 14-17, 14-21
 - bearer capability 14-11
 - cadences
 - modifying 14-20
 - call divert 14-13, 14-17, 14-21
 - call forwarding 14-9
 - call logging 14-12
 - call priority 14-2, 14-10
 - call transfer 14-9
 - call waiting 14-8
- calls 14-6
 - displaying 14-26
 - external, answering 14-7
 - external, clearing 14-7
 - external, making 14-6
- commands
 - CREATE PBX EXTENSION 14-13
 - CREATE PBX GROUP 14-16
 - DESTROY PBX EXTENSION 14-17
 - DESTROY PBX GROUP 14-18
 - DISABLE PBX DEBUG 14-18
 - ENABLE PBX DEBUG 14-19
 - SET PBX 14-20
 - SET PBX EXTENSION 14-21
 - SET PBX GROUP 14-24
 - SHOW PBX 14-25
 - SHOW PBX CALL 14-26
 - SHOW PBX EXTENSION 14-27
 - SHOW PBX GROUP 14-29
- conference calling 14-8
- configuration
 - displaying 14-25
- country
 - modifying 14-20
- country setting 14-20
- creating
 - extensions 14-13
 - groups 14-16
- DDI support 14-2, 14-9
- debugging
 - disabling 14-18
 - enabling 14-19
- destroying
 - extensions 14-17
 - groups 14-18
- displaying
 - calls 14-26
 - extensions 14-27
 - general configuration 14-25
 - groups 14-29
- enbloc dialling 14-2, 14-12, 14-20
- extensions 14-3
 - creating 14-13



- destroying 14-17
- displaying 14-27
- modifying 14-21
- external calls 14-2
- general configuration
 - displaying 14-25
- groups 14-4
 - creating 14-16
 - destroying 14-18
 - displaying 14-29
 - modifying 14-24
- hunting 14-2, 14-16, 14-18, 14-24
- Least Cost Routing (LCR)
 - tieline dialling 14-20
- logging 14-2, 14-12
- modifying
 - cadences 14-20
 - country 14-20
 - extensions 14-21
 - groups 14-24
 - prefixes 14-20
 - territory 14-20
- MSN support 14-2, 14-9
- numbers 14-4
- operator dial 14-20
- overlap dialling 14-2, 14-12, 14-20
- ports 14-2, 14-13, 14-17, 14-21
- prefixes 14-20
 - modifying 14-20
- speech encoding 14-12
- territory
 - modifying 14-20
- tieline dialling 14-20
- tone cadencing 14-12, 14-20
- tone suppression 14-2, 14-12
- tones 14-4
 - bell 14-4
 - external dial 14-5
 - unavailable 14-5
- VOX ports 14-2
- Telnet 1-12, 7-5 to 7-6, 7-11
 - for remote management 1-12
 - from the router 1-11
- TELNET command 7-11
- templates, PPP interfaces 3-9
 - creating 3-36
 - debugging 3-43, 3-46
 - destroying 3-42
 - displaying 3-80
 - modifying 3-54
- terminal server 7-1 to 7-12
 - accessing Telnet hosts 7-5 to 7-6, 7-11
 - asynchronous ports 2-7, 2-14 to 2-15, 2-17 to 2-19, 2-33
 - auto-bauding 2-11
 - command line editing 7-4
 - command line recall 7-4
 - commands 7-6
 - DISABLE PORT 2-14
 - ENABLE PORT 2-15
 - PURGE PORT 2-15
 - RESET PORT 2-17
 - RESET PORT COUNTERS 2-17
 - RESET PORT HISTORY 2-18
 - SET PORT 2-19
 - SET TELNET 7-6
 - SET TTY 7-7
 - SHOW PORT 2-33
 - SHOW TTY 7-8
 - TELNET 7-11
 - configuring asynchronous ports 2-7, 2-14 to 2-15, 2-17 to 2-19
 - configuring Telnet 7-6
 - configuring TTY devices 7-7
 - creating nicknames for Telnet hosts 7-5
 - default asynchronous port characteristics 2-8
 - disabling ports 2-14
 - displaying
 - TTY devices 7-8
 - displaying asynchronous port characteristics 2-9, 2-33
 - enabling ports 2-15
 - history 2-18, 7-4
 - ports 2-7, 2-14 to 2-15, 2-17, 2-19
 - purging port configuration 2-15
 - resetting command history 2-18
 - resetting port counters 2-17
 - resetting ports 2-17
 - setting port characteristics 2-7, 2-19
 - Telnet 7-5, 7-11
 - TTY device 7-2
 - configuring 7-7
 - displaying 7-8
- territory
 - displaying 1-78
 - setting 1-56
- Testing
 - commands
 - DISABLE TEST INTERFACE 9-8
 - ENABLE TEST INTERFACE 9-9
 - RESET TEST INTERFACE 9-10
 - SHOW TEST 9-10
 - disabling tests 9-8
 - displaying test results 9-10
 - enabling tests 9-9
 - expansion options 9-1 to 9-12
 - hardware 9-1 to 9-12
 - interfaces 9-1 to 9-12
 - loopback plugs 9-3
 - resetting tests 9-10
- text editor 1-18, 1-37
- textual notation 16-4
- tieline dialling, LCR 14-20
- time
 - displaying 1-80
 - setting 1-56
 - triggers 10-2
- Time Division Multiplexing (TDM) 11-1 to 11-7
 - adding time slots 11-3
 - commands
 - ADD TDM 11-3



- CREATE TDM 11-4
- DELETE TDM 11-5
- DESTROY TDM 11-5
- PURGE TDM 11-6
- SHOW TDM 11-6
- configuration example 11-2
- creating TDM groups 11-4
- deleting time slots 11-5
- destroying TDM groups 11-5
- displaying TDM groups 11-6
- displaying time slots 11-6
- E1/T1 11-2
- example configuration 11-2
- mixed mode 11-2
- Primary Rate ISDN interfaces 11-2
- purging all TDM groups 11-6
- TDM group
 - adding time slots 11-3
 - creating 11-4
 - creating PPP interfaces over 11-3 to 11-4
 - deleting time slots 11-5
 - destroying 11-5
 - displaying 11-6
 - purging all 11-6
- time slots 11-2
 - adding 11-3
 - deleting 11-5
 - displaying 11-6
- timers
 - Point-to-Point Protocol (PPP) 3-19
- TRACE command 6-146
- trace route 6-25, 6-104, 6-143, 6-145 to 6-146
- traffic filters 6-21
- Transmission group, MIB-II MIB C-12 to C-13
- trap PDU 16-6
- trap, SNMP protocol 16-7
- traps MIB object C-4
- triggers 10-1 to 10-21
 - actions 10-2
 - adding to triggers 10-3, 10-6 to 10-7
 - deleting from triggers 10-3, 10-11
 - displaying 10-16
 - activation 10-2 to 10-3, 10-5
 - automated command execution 10-2
 - commands
 - ACTIVATE TRIGGER 10-5
 - ADD TRIGGER 10-6
 - CREATE TRIGGER 10-7
 - DELETE TRIGGER 10-11
 - DESTROY TRIGGER 10-11
 - DISABLE TRIGGER 10-12
 - ENABLE TRIGGER 10-12
 - PURGE TRIGGER 10-13
 - SET TRIGGER 10-13
 - SHOW TRIGGER 10-16
 - CPU 10-7, 10-13
 - creating 10-3, 10-7
 - deleting 10-11
 - destroying
 - triggers 10-3, 10-11

- disabling 10-3, 10-12
- displaying
 - triggers 10-3, 10-16
- enabling 10-3, 10-12
- firewall 10-7, 10-13
- interface 10-7, 10-13
- link 10-7, 10-13
- manual activation 10-3, 10-5
- memory 10-7, 10-13
- modifying 10-3, 10-13
- output 10-2
- periodic 10-2, 10-7, 10-13
- purging 10-13
- reboot 10-2, 10-7, 10-13
- response to events 10-2
- time 10-2, 10-7, 10-13
- timed command execution 10-2
- triggers
 - destroying 10-3
 - types 10-2, 10-7, 10-13
- trusted routers 6-61, 6-69, 6-136

U

- U interfaces, Basic Rate Access 4-6
- UAF. See User Authentication Facility
- UDP group, MIB-II MIB C-11
- Universal Coordinated Time (UTC) 12-4
- UPLOAD command 1-82
- User Authentication Database 1-7 to 1-8
 - adding users 1-8, 1-28
 - counters 1-11
 - deleting users 1-8, 1-33
 - disabling users 1-35
 - displaying users 1-9, 1-80
 - enabling users 1-41
 - logging in and out 1-10, 1-44 to 1-45
 - modifying users 1-8, 1-57
 - passwords 1-9
 - purging users 1-46
 - recovering passwords 1-11
 - resetting users 1-48
 - security 1-9
- User Authentication Facility 1-7
 - asynchronous port security 1-11
 - semipermanent manager port 1-12
 - Telnetting from the router 1-11
- User Authentication Facility (UAF)
 - database 3-15
- USER privilege 1-3
- UTC. See Universal Coordinated Time (UTC)

V

- Van Jacobson header compression 8-3
- view, SNMP MIB 16-8

W

- WAIT command 13-8, 13-12
- web server. See HTTP server
- wildcard characters
 - in file names 1-16

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

write-only, MIB object 16-5

WWW server. See HTTP server

X

X.25 5-1 to 5-35

activating MIOX circuits 5-9

adding

call parameters 5-11

MIOX circuits 5-10, 5-19

addresses

DTE 5-4

call parameters

adding 5-11

configuring 5-7

deleting 5-15

displaying 5-34

modifying 5-22

circuit 5-2

commands 5-8

ACTIVATE MIOX CIRCUIT 5-9

ADD MIOX CIRCUIT 5-10, 5-19

ADD X25T CPAR 5-11

CREATE X25T 5-12

DEACTIVATE MIOX CIRCUIT 5-14

DELETE MIOX CIRCUIT 5-15

DELETE X25T CPAR 5-15

DESTROY X25T 5-16

DISABLE MIOX CIRCUIT 5-16

ENABLE MIOX CIRCUIT 5-17

RESET X25T 5-18

SET MIOX 5-18

SET MIOX CIRCUIT 5-19

SET X25T 5-21

SET X25T CPAR 5-22

SHOW MIOX 5-23

SHOW MIOX CIRCUIT 5-26

SHOW MIOX COUNT 5-24

SHOW X25T 5-30

SHOW X25T CPAR 5-34

configuring

call parameters 5-7

DTE 5-6

permanent virtual circuit (PVC) 5-7

creating

X.25 DTE interfaces 5-12

Data Communication Equipment (DCE) 5-2

Data Terminal Equipment (DTE) 5-2

DCE 5-2

deactivating MIOX circuits 5-14

deleting

call parameters 5-15

MIOX circuits 5-15

destroying

X.25 DTE interfaces 5-16

disabling

MIOX circuits 5-16

displaying

call parameters 5-34

MIOX 5-23

MIOX circuits 5-26

MIOX counters 5-24

X.25 DTE counters 5-30

X.25 DTE interfaces 5-30

DTE 5-2

configuring 5-6

DTE addresses 5-4

DTE counters

displaying 5-30

DTE interfaces

creating 5-12

destroying 5-16

displaying 5-30

modifying 5-21

reseting 5-18

DTE mode 5-3

enabling

MIOX circuits 5-17

encapsulations 5-4

MIOX

displaying 5-23

modifying 5-18

MIOX circuits 5-4

activating 5-9

adding 5-10, 5-19

deactivating 5-14

deleting 5-15

disabling 5-16

displaying 5-26

enabling 5-17

modifying 5-19

MIOX counters

displaying 5-24

modifying

call parameters 5-22

MIOX 5-18

MIOX circuits 5-19

X.25 DTE interfaces 5-21

over ISDN 4-12, 4-51 to 4-52, 4-55

packet assembler/disassembler (PAD) 5-2

permanent virtual circuit (PVC) 5-3

configuring 5-7

permanent virtual circuits (PVC) 5-2

reseting

X.25 DTE interfaces 5-18

standards 5-2

switched virtual circuit (SVC) 5-2 to 5-3

virtual circuit 5-2